SemEval 2024

The 18th International Workshop on Semantic Evaluation (SemEval-2024)

Proceedings of the Workshop

June 20-21, 2024

The SemEval organizers gratefully acknowledge the support from the following organizations.

Gold

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 979-8-89176-107-0

Introduction

The Semantic Evaluation (SemEval) workshops focus on the evaluation and comparison of systems that analyze diverse semantic phenomena in text, with the aim of extending the current state of the art in semantic analysis and creating high quality annotated datasets in a range of increasingly challenging problems in natural language semantics. SemEval provides an exciting forum for researchers to propose challenging research problems in semantics and to build systems/techniques to address such research problems.

SemEval-2024 is the eighteenth workshop in the series of International Workshops on Semantic Evaluation. The workshop began in 1998 and was originally known as SensEval and focused on word sense disambiguation.
In 2007, the workshop was renamed SemEval, and evolved to include semantic tasks beyond word sense disambiguation. Starting in 2012, SemEval has been organized every year. The tasks for the next iteration of the workshop, SemEval-2025 (https://semeval.github.io/SemEval2025/), are underway.

SemEval-2024 is co-located (hybrid) with the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2024). SemEval-2024 will be held in Mexico City, Mexico and it includes the following 10 tasks:

- Semantic Relations
  - Task 1: Semantic Textual Relatedness for African and Asian Languages
  - Task 2: Safe Biomedical Natural Language Inference for Clinical Trials
- Discourse and Argumentation
  - Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations
  - Task 4: Multilingual Detection of Persuasion Techniques in Memes
  - Task 5: Argument Reasoning in Civil Procedure
- LLM Capabilities
  - Task 6: SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes
  - Task 7: NumEval: Numeral-Aware Language Understanding and Generation
  - Task 8: Multidomain, Multimodel and Multilingual Machine-Generated Text Detection
- Knowledge Representation and Reasoning
  - Task 9: BRAINTEASER: A Novel Task Defying Common Sense
  - Task 10: Emotion Discovery and Reasoning its Flip in Conversation

This volume contains both the task description papers (10), that describe each of the above tasks, and the system description papers (279) that present the systems that participated in the tasks.
In addition, SemEval-2024 features two awards, one for the organizers of a task and one for a team participating in a task. The Best Task award recognizes a task that stands out for making an important intellectual contribution to empirical computational semantics, as demonstrated by a creative, interesting, and scientifically rigorous dataset and evaluation design, and a well-written task overview paper. The three Best System Description Paper awards recognize a system description paper (written by a team participating in one of the tasks) that advances our understanding of a problem and available solutions with respect to a task. It does not need to be the highest scoring system in the task, but it should have a strong analysis component in the evaluation, as well as a clear and reproducible description of the problem, algorithms, and methodology.

We are grateful to the task organizers for their dedication in carrying out ten very successful tasks and to the large number of participants whose enthusiastic participation has made SemEval 2024 a successful event. We also appreciate the efforts of the task organizers and participants who reviewed the paper submissions. These proceedings have greatly benefited from their detailed and thoughtful feedback. Finally, we also thank the members of the program committee who reviewed the submitted task proposals and helped us to select this exciting set of tasks, the NAACL 2024 conference organizers for their support, and the ACL Special Interest Group on the Lexicon (SIGLEX) for sponsoring and supporting this event.

Atul Kr. Ojha, A. Seza Doğruöz, Harish Tayyar Madabushi, Giovanni Da San Martino, Sara Rosenthal, and Aiala Rosá (SemEval-2024 Organizers and Co-Chairs)

Organizing Committee

SemEval Chairs

Atul Kr. Ojha, Data Science Institute, Unit for Linguistic Data, University of Galway
A. Seza Doğruöz, Universiteit Gent
Harish Tayyar Madabushi, University of Bath
Giovanni Da San Martino, University of Padova
Sara Rosenthal, IBM Research
Aiala Rosá, Instituto de Computación, Facultad de Ingeniería, Universidad de la República

Program Committee

Alessandro Raganato, University of Milano-Bicocca
Alexandra Uma, Queen Mary University of London
Ali Zeynali, College of Information and Computer Science, University of Massachusetts at Amherst
Amitava Das, University of of South Carolina
Andrey Kutuzov, University of Oslo
Anjie Fang, Amazon
Besnik Fetahu, Amazon
Boyuan Zheng, Ohio State University, Columbus
Carla Perez-Almendros, Cardiff University
Chenxi Whitehouse, City, University of London
Chico Q Camargo, University of Exeter
Chung-chi Chen, National Institute of Advanced Industrial Science and Technology
Corey A. Harper, Elsevier Labs
Cristiano Chesi, Istituto Universitario di Studi Superiori
Cunxiang Wang, Westlake University
David Jurgens, University of Michigan - Ann Arbor
Dimitar Dimitrov, University of Sofia "St. Kliment Ohridski"
Egoitz Laparra, University of Arizona
Elaine Zosa, SiloGen
Fanfan Wang, Nanjing University of Science and Technology
Filip Ilievski, Vrije Universiteit Amsterdam
Firoj Alam, Qatar Computing Research Institute, HBKU
Francesca Gasparini, University of Milano - Bicocca
Francesco Barbieri, Snap Inc.
Gavin Abercrombie, Heriot-Watt University
Gustavo H. Paetzold, Universidade Tecnologica Federal do Parana
Henning Wachsmuth, Leibniz Universitat Hannover
Ibrahim Abu Farha, University of Sheffield
Ibrahim Said Ahmad, Northeastern University
Idris Abdulmumin, Ahmadu Bello University, Zaria
Ivan Habernal, Paderborn University
Jakub Piskorski, Institute of Computer Science, Polish Academy of Science
Jeffrey S. Sorensen, Google
Jennifer D'Souza, TIB Hannover
Jeremy Barnes, University of the Basque Country
Jianfei Yu, Nanjing University of Science and Technology
Jingxuan Tu, Brandeis University

John Pavlopoulos, Stockholm University
Jonibek Mansurov, MBZUAI
Jose Camacho-Collados, Cardiff University
Kaixin Ma, Carnegie Mellon University
Lena Held, Technische UniversitÃ¤t Darmstadt
Mael Jullien, university of Manchester
Maik Frobe, Martin-Luther Universitat Halle-Wittenberg
Marcos Garcia, Universidade de Santiago de Compostela
Marko Robnik-Sikonja, University of Ljubljana
Matthew Purver, Queen Mary University of London
Matthew Shardlow, The Manchester Metropolitan University
Maximilian Heinrich, Bauhaus Universitat Weimar
Md. Shad Akhtar, Indraprastha Institute of Information Technology, Delhi
Meriem Beloucif, Uppsala University
Mohamed Abdalla, University of Toronto
Mohammad Taher Pilehvar, Tehran Institute for Advanced Studies
Nailia Mirzakhmedova, Bauhaus Universitat Weimar
Nedjma Ousidhoum, Cardiff University
Nicolas Stefanovitch, European Commission
Osama Mohammed Afzal, MBZUAI
Paolo Rosso, Universitat Politecnica de Valencia
Preslav Nakov, Mohamed bin Zayed University of Artificial Intelligence
Raul Vazquez, University of Helsinki
Roberto Zamparelli, University of Trento
Rui Xia, Nanjing University of Science and Technology
Saif Mohammad, NRC
Sanchit Ahuja, Microsoft Research
Seid Muhie Yimam, Universität Hamburg
Shervin Malmasi, Amazon
Shivani Kumar, Indraprastha Institute of Information Technology
Somin Wadhwa, Northeastern University
Steven Bethard, University of Arizona
Steven Schockaert, Cardiff University
Tanmoy Chakraborty, IIT Delhi
Teemu Vahtola, University of Helsinki
Timothee Mickus, University of Helsinki
Timothy A Miller, Harvard University
Tommaso Fornaciari, Italian National Police
Vivek Khetan, Accenture Labs
Vladimir Araujo, KU Leuven
Xi Chen, Department of Computer Science, University of Massachusetts at Amherst
Yifan Jiang, Information Sciences Institute,University of Southern California
Yuxia Wang, The University of Melbourne
Zhiyu Chen, Amazon

Keynote Talk: Beyond Single Scores: Transparent Evaluation through Fine-Grained Error Detection and Uncertainty Quantification

André F. T. Martins
Instituto Superior Tecnico, Senior Researcher at the Instituto de Telecomunicacoes, and VP of AI Research at Unbabel in Lisbon, Portugal

Abstract: Automatic evaluation metrics are key to drive progress in NLP. We use them to compare systems and decide which models to deploy, to understand the strengths and weaknesses of each model, and to help practitioners overcome existing failure modes. In this talk, I will discuss evaluation of machine translation quality. Today, lexical-based metrics (such as BLEU or ChrF) are being replaced by learned neural-based metrics, such as COMET and BLEURT, which exhibit much better correlation with human judgments. However, these metrics provide a single sentence-level score, offering little insight into translation errors (e.g., what are the errors and what is their severity). Can we do better? I will start by presenting xCOMET, an open-source learned metric which integrates both sentence-level evaluation and error span detection capabilities, exhibiting state-of-the-art performance across all types of evaluation (sentence-level, system-level, and error span detection). Moreover, it does so while highlighting and categorizing error spans, thus enriching the quality assessment. Then, I will discuss recent approaches that endow evaluation metrics with uncertainty quantification capabilities, using techniques such as Monte Carlo dropout, deep ensembles, heteroscedastic regression, quantile regression, and conformal prediction. Finally, I will present Tower, an open multilingual LLM for translation-related tasks. We perform continued pretraining on a multilingual mixture of monolingual and parallel data, creating TowerBase, followed by finetuning on instructions relevant for translation processes, creating TowerInstruct. The final model surpasses open alternatives on several tasks relevant to translation workflows and is competitive with general-purpose closed LLMs. To facilitate future research, we release the Tower models, our specialization dataset, an evaluation framework for LLMs focusing on the translation ecosystem, and a collection of model generations, including ours, on our benchmark.

Bio: André F. T. Martins is an Associate Professor at Instituto Superior Técnico, Senior Researcher at the Instituto de Telecomunicações, and VP of AI Research at Unbabel in Lisbon, Portugal. I also do scientific consulting for Priberam Labs. I work on natural language processing and machine learning.

Until 2012, André was a PhD student in the joint CMU-Portugal program in Language Technologies, at Carnegie Mellon University and Instituto Superior Técnico. His advisors were Mario Figueiredo, Noah Smith, Pedro Aguiar and Eric Xing.

Table of Contents

Program

Thursday, June 20, 2024

09:10 - 09:25 *Welcome and Introduction to SemEval*

09:30 - 10:30 *Invited Talk 1, Shared with *SEM*

10:30 - 11:00 *Coffee Break*

11:00 - 12:30 *Oral Session-I*

SemEval Task 1: Semantic Textual Relatedness for African and Asian Languages
Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif and Christine De Kock

SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials
Mael Jullien, Marco Valentino and André Freitas

SemEval-2024 Task 3: Multimodal Emotion Cause Analysis in Conversations
Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu and Erik Cambria

SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes
Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov and Giovanni Da San Martino

SemEval-2024 Task 5: Argument Reasoning in Civil Procedure
Lena Held and Ivan Habernal

SemEval-2024 Task 6: SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes
Timothee Mickus, Elaine Zosa, Raul Vazquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato and Marianna Apidianaki

12:30 - 14:00 *Lunch*

14:00 - 15:00 *Oral Session-II*

SemEval-2024 Task 7: Numeral-Aware Language Understanding and Generation
Chung-chi Chen, Jian-tao Huang, Hen-hsen Huang, Hiroya Takamura and Hsin-hsi Chen

Thursday, June 20, 2024 (continued)

SemEval-2024 Task 8: Multidomain, Multimodel and Multilingual Machine-Generated Text Detection
Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti and Thomas Arnold

SemEval-2024 Task 9: BRAINTEASER: A Novel Task Defying Common Sense
Yifan Jiang, Filip Ilievski and Kaixin Ma

SemEval 2024 - Task 10: Emotion Discovery and Reasoning its Flip in Conversation (EDiReF)
Shivani Kumar, Md. Shad Akhtar, Erik Cambria and Tanmoy Chakraborty

15:00 - 15:30 *Best System Paper's Presentations*

15:30 - 16:00 *Coffee Break*

16:00 - 17:30 *Poster Session I: System Description Papers (local and online)*

Friday, June 21, 2024

09:30 - 10:30 *Invited Talk: Beyond Single Scores: Transparent Evaluation through Fine-Grained Error Detection and Uncertainty Quantification (André F. T. Martins)*

10:30 - 11:00 *Coffee Break*

11:00 - 12:30 *Poster Session II: System Description Papers (online)*

12:30 - 14:00 *Lunch*

14:00 - 15:30 *Poster Session III: System Description Papers (in presence)*

15:30 - 16:00 *Coffee Break*

16:00 - 16:45 *Best Paper Awards and Concluding Remarks*

CUNLP at SemEval-2024 Task 8: Classify Human and AI Generated Text

Aggarwal Pranjal, Sachdeva Deepanshu

University of Colorado Boulder

(pranjal.aggarwal, deepanshu.sachdeva)@colorado.edu

Abstract

This task is a sub-part of SemEval-2024 competition which aims to classify AI vs Human Generated Text. In this paper we have experimented on an approach to automatically classify an artificially generated text and a human written text. With the advent of generative models like GPT-3.5 and GPT-4 it has become increasingly necessary to classify between the two texts due to various applications like detecting plagiarism and in tasks like fake news detection that can heavily impact real world problems, for instance stock manipulation through AI generated news articles. To achieve this, we start by using some basic models like Logistic Regression and move our way up to more complex models like transformers and GPTs for classification. This is a binary classification task where the label 1 represents AI generated text and 0 represents human generated text. The dataset was given in JSON style format which was converted to comma separated file (CSV) for better processing using the pandas library in Python as CSV files provides more readability than JSON format files. Approaches like Bagging Classifier and Voting classifier were also used.

1 Introduction

We perform Subtask A of the Task 8 [1] from the International Workshop on Semantic Evaluation: SemEval 2024[†] which stated - *Multidomain, Multimodal and Multilingual Machine-Generated Text Detection*. In this subtask we perform Monolingual (English in this case) classification for AI generated vs Human written texts.

This Binary classification task has utmost utility in real world scenarios like - content moderation on social media platforms, fake news detection that can impact organizations financially and people emotionally, detecting spam messages in email or communication channels like Slack.

Another application can be used in healthcare chatbots to make sure that a person is talking to a person as this kind of task needs human speciality. Product reviews classification - i.e., detecting whether an organization has human written reviews, or they had them generated through AI to rank their product higher up in the chain.

To perform this task, we use a series of techniques including manual feature engineering for supervised learning techniques like logistic regression and Bagging Classifier as well as more complex techniques like Neural Networks and attention mechanism with transformers. We used supervised learning as well like K-Nearest Neighbours. The best approach found was a combination of transformers [2] with hand engineered features like Coherence [3] of a text, Complexity, length and emoji count. The accuracy and performance of these experiments are discussed in the later sections.

In our experiments we found that some features were very influential like length of a text, vocabulary used in the text and coherence of a text. Other features like complexity of the text had less weightage and were thus, not used in all experiments. Even though transformers gave us the best accuracy we also used some other approaches that were competitive as well.

We also had some limitations in the usage of computing resources where one of our approaches that combines TF-IDF vector along with transformers uses over 50 GB of RAM that exceeds the amount of any available computing resource available to us.

2 Background

Dataset - The dataset that was used was provided by SemEval that is an extension of the M4 dataset [4]. which had approximately 133551 data points in the training set and the dev set contained 5000 samples. The dataset contained texts from various sources

[†] https://semeval.github.io/SemEval2024/tasks

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1–6

including Wikipedia, Reddit, WikiHow, and PeerRead for English texts. The AI generated text was curated from Generative models like ChatGPT, Cohere, Dolly v-2 and Bloomz. After analysing the data, we found that the dev set data only included the data points from Bloomz and there were none of Bloomz model's generated texts in the training set. This was meant to test the real-life situation where a new generative model can come into picture when our model would not have seen that generative model's pattern.

An exploratory data analysis of the text, gave the following interesting observations:

1. The training set has a total vocabulary size of *2616365* in which there were around *328491* words that were only used by the AI generated texts..
2. The total number of unique words used by AI generated text was *581888* as compared to that used by Humans which was *2034477*. This data suggests that AI used a lot of repetitive words as compared to humans.
3. The average number of tokens used in a sentence generated by humans were - *283* as compared to AI which used only *155*.

Figure 1: AI Corpus Word Cloud

Key terms: Step, Part, Overall, S, One, Make

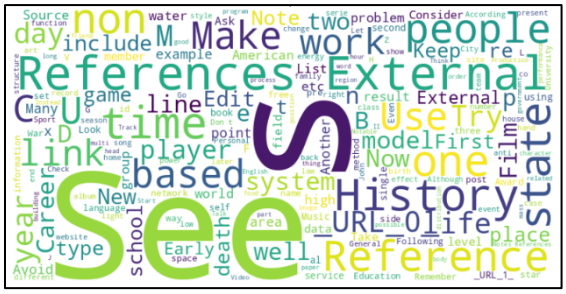


Figure 2: Human Corpus Word Cloud

Key terms: See, S, References, External, History

We also explored some aspects of sentence structure like coherence, complexity and length of the sentences. We used these features along with the TF-IDF vectors as an input to the Logistic Regression model, Bagging Classifier and K-Nearest Neighbours achieving a max training set accuracy score of *0.91* and *0.61* on dev set using Bagging Classifier. We also used a voting classifier which performed better, achieving an accuracy score of *0.68 on the dev set* using the above-mentioned models. Later we used transformers (BERT) with combination of the above-mentioned models in the Voting classifier. There were two more approaches where we tried topic modelling and feature repetition which yielded better results.

One thing to note here is that when a certain text is generated by AI it contains some sort of template or pattern around it. So, to use that we tried unsupervised learning to make possible clusters of the texts, to identify which class of template the text might belong to. This approach included the use of the K-means clustering method, which reported a dev set accuracy of 0.57

Heather et al. [5] mentions the use of simple machine learning techniques with great accuracy. Ahmed et al. [6] compared different methodologies and tools and how each of them perform on unseen data.

In any of the literature TF-IDF was not used along with any other features, and we experimented by including these features in our approach along with topic modelling setup that was novel.

3 System Overview

Text Classification even though an already accomplished task becomes challenging even for state-of-the-art models like Transformers. In this task the adaptability of GPT makes it even more challenging to differentiate between the two types of texts. Also, as AI progresses to understand human emotions [7] and behaviour it is expected from the model to generate texts i.e. convey its thought in a more human centric manner. We aim to tackle the same starting with the standard machine learning algorithms and then moving on to much more complex models like attention based transformer models, example - BERT [8], RoBERTa [9] among others.

We describe below in detail the specification used along with each approach and mention its accuracy and experimental setup.

For this task, we have used TF-IDF vectorization technique. Along with that we also analysed text structures and engineered 3 main features related to the task at hand. These were Complexity of the sentence, Coherence of a text and length of text (tokenization).

These features were used by the algorithms described below and are described in the next section in detail.

1. **Standard ML Algorithms with TF-IDF:** As this is a binary classification task, we start by using logistic regression. We used TF-IDF vectors as input to this. As discussed earlier, human text used a wide range of vocabulary with an average length of around 283 words, AI generated text used a smaller vocabulary set and the average sentence length was around 155 words. There were a lot of words that were not used in human Corpus (around 3.5 lakhs), so we used TF IDF Vector as the input to various machine learning models such as logistic regression, bagging classifier and unsupervised learning technique K-Nearest Neighbours.
2. **BERT:** BERT or Bidirectional Encoder Representations from Transformers uses an attention mechanism to capture the essential information for a given task. We used the BERT based uncased model as a baseline to compare the performance of our algorithms. Variations of BERT like RoBERTa, XLM-RoBERTa [10] were also used along with experimentation with our manually engineered features (with and without repetition) achieving a dev set accuracy of 0.66. Repetition of features is described in the experimental setup in more detail.
3. **Transformers with Features:** Features like Coherence and length of text were used in addition to the tokens that were passed in the transformer models. These were passed in the form of a list followed by tokens inputted into the transformers model. These features though could be imagined to be captured by the model itself but being complex features, it makes more sense to extract these features from the models specifically trained for this purpose. This helped us enhance the efficiency and performance of our models. Since these features were less in number, to increase their effect on the output, the features were repeated, and the repetition was treated as a hyperparameter, this value was randomly assigned in the range from 200 to 300.
4. **Transformers with TF-IDF and SVD:** Since TF-IDF is a feature that proves to be useful in trivial machine learning algorithms like logistic regression, we experimented to use it with much more complex models like state of the art - transformers. Since, using transformers itself is computationally expensive, along with TF-IDF the computational complexity increases exponentially, requiring over 50 GB of CPU memory to prepare the input tensor. Due to the lack of such computational resources, we relied on dimensionality reduction algorithms such as Singular Value Decomposition (SVD). After experimentation over 1 epoch, although requires more research, were appreciable.
5. **Topic Modelling with Transformers:** A common trait in a generative model is that the output follows from a particular prompt. That means that every text generated by the AI model can be segregated into a certain topic. So, we aim to use topic modelling as a feature to the input tensor while classifying AI and human generated text. As every human has a certain way of writing, similarly every AI model can be said to have a way of generating text. So here we approach this method by first using an unsupervised learning technique such as K-Means clustering that separates text into a certain number of clusters. This number again is a hyperparameter set to 100 in this experiment that can be set by the experimenter. After that, the output of this model i.e., the cluster number is fed into higher order models such as transformers to gain better results and an accuracy of 0.56 was achieved on the dev set.

4 Experimental Setup

Various experiments were performed on the given dataset. The train-test split for all the experiments was kept the same to the ratio of 80:20. This split comes from the training data itself and the dev set was kept unseen from the model during the training phase. The best results on the dev set after hyperparameter tuning are logged in the results section of the paper. In this section we discuss the following:

1. **Performance Metrics:** We used micro-F1 and macro-F1 scores as well as accuracy itself to measure the performance of the model across various algorithms. We also monitored precision and recall and observed lower recall rates across the models. This means that the algorithms are biased towards classifying the output as AI generated text. This recall was later used as a weightage in the voting classifier.
2. **Feature Engineering:** We used different features as input to models, like:
   a. Complexity of a Sentence: Using the 'textstat' module in python we calculated

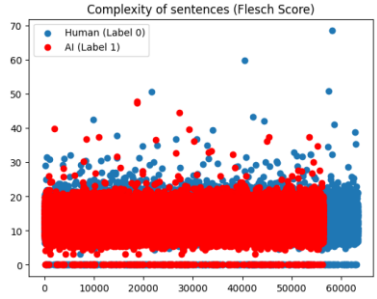


Figure 3: Complexity

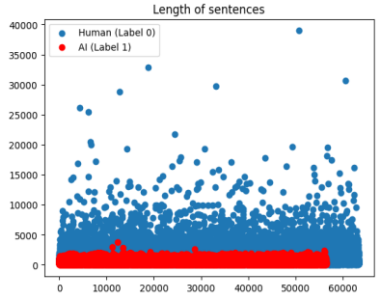


Figure 4: Length

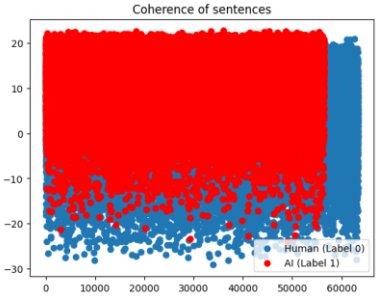


Figure 5: Coherence

the Flesch Score that indicates the readability of a sentence in the range of 0 to 100, with 0 being most confusing and 100 being very easy to understand.

b. Length of a Sentence: Observing the significant difference between the average length of text between AI generated text and Human Generated Text, we decided to use it as a feature to our ML algorithms. The average length of text in AI generated text was noted to be 155, however it was 283 for human generated text. The length of the sentence was calculated by first removing the stop words using the NLTK library, followed by lemmatization and then counting the number of tokens after the operation.

c. Coherence of Text: It is the measure of transitions in a text along with smoothness and logical flow. The coherence of text is an important feature, we observed that a human generated text was more coherent than AI generated. Coherence of the text was calculated using the SGNLP library in Python.

The comparison of AI generated text and human written text on the above features are shown in figures 3, 4, and 5, respectively. These features are referred as "sentence features" from now on in the paper.

3. **Loss Function:** The loss function used for logistic regression is the binary cross entropy loss. The same loss function has been used in Transformers as well.
4. **Optimizer:** Different optimization algorithms including Adam, AdaGrad and RMSProp were used during experimentation and the best performance was shown by Adam optimizer.
5. **Computational Resources:** Kaggle and Google Colab were used interchangeably for experimentation. However, since GPU was a requirement and the average time for experimentation for 1 epoch exceeded over 4 hours, multiple experiments were run on the Kaggle platform on a T4x2 GPU accelerator, this setup was exclusively used for transformers-based experiments. For experiments on machine learning algorithms, 12 GB CPU RAM was sufficient and hence Google Colab was used.
6. **Hyperparameter Tuning:** There were several hyper-parameters that required tuning over the course of this experiment, most of the hyper-parameter tuning was done in transformers with learning rate, weight decay, epochs and optimizer choice. Grid search was used to obtain the most optimal values of hyper-parameters. Other custom hyperparameters were also involved such as the number of repetition of features, d-dimensionality reduction in experimentation of TF-IDF with transformers and the number of topic models to be included as a feature in addition to transformers.

5 Results

We observed that the model combined with the attention mechanism of transformers with TF-IDF vectors provides is with the best results. However, it should be noted that the dimensionality of the vectors has been significantly reduced due to its computational complexity and thus is bound to affect the accuracy. The results mentioned in the below table (Table 1) are the optimal results obtained after repeated experimentation over different optimizers, epochs and weight decay rates. Some parameters have not been mentioned in the table, as the standard grid search can be reimplemented if there is a need for replication. As evident from the table, the best results were obtained when we used the XLM-RoBERTa model along with TF-IDF features and the sentence features (complexity, length and coherence).

Model	Accuracy	Epoch	Precision	F1
Logistic Regression	0.49	-	0.48	0.31
Bagging Classifier	0.57	-	0.55	0.42
Voting Classifier (LR, Bagging, KNN)	0.57	-	0.55	0.42
BERT (with Sentence Features)	0.72	1	0.94	0.71
RoBERTa (with Features)	0.77	2	0.96	0.73
XLM-RoBERTa (with TF-IDF and Sentence Features)	0.78	2	0.97	0.74

Table 1: Performance of different models

6 Conclusion

This Binary Classification task of predicting the mode of text generation is non-trivial in the aspect that as the generative models are largely trained on human generated text, they have learned to write more like humans and thus this becomes a challenging task. However, using proper means and computational methods, it is possible to segregate them using conventional feature extraction techniques combined with self-attention mechanism of transformers as seen in the experiments. We aim to use the topic modelling approach combined with TF-IDF and transformers further in the future that might yield promising results.

References

[1] J. M. P. I. J. S. A. S. A. T. O. M. A. T. M. G. P. T. A. C. W. A. F. A. N. H. I. G. P. N. Yuxia Wang, "SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection," *Proceedings of the 18th International Workshop on Semantic Evaluation,* vol. SemEval 2024, June 2024.

[2] N. S. N. P. J. U. L. J. A. N. G. L. K. I. P. Ashish Vaswani, "Attention Is All You Need," *CoRR,* 2017.

[3] Y. L. Y. Z. Z. Z. Baiyun Cui, "Text Coherence Analysis Based on Deep Neural Network," *CoRR,* 2017.

[4] J. M. P. I. J. S. A. S. A. T. C. W. O. M. A. T. M. T. S. T. A. A. F. A. N. H. I. G. P. N. Yuxia Wang, "M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection," *arXiv:2305.14902,* 2023.

[5] A. E. C. M. I. R. J. D. H. Heather Desaire, "Distinguishing academic science writing from humans or ChatGPT with over 99% accuracy using off-the-shelf machine learning tools," *Cell Reports Physical Science,* vol. 4, no. 6, 2023.

[6] K. E. S. A. Ahmed M. Elkhatat, "Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text," *International Journal for Educational Integrity,* vol. 19, no. 17, 2023.

[7] H. N.-M. W. C. Francisca Adoma Acheampong, "Transformer models for text-based emotion detection: a review of BERT-based approaches," *Artificial Intelligence Review,* vol. 54, no. 8, pp. 5789-5829, 2021.

[8] M.-W. C. K. L. K. T. Jacob Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *CoRR,* 2018.

[9] M. O. N. G. J. D. M. J. D. C. O. L. M. L. L. Z. V. S. Yinhan Liu, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *CoRR,* 2019.

[10] K. K. N. G. V. C. G. W. F. G. E. G. M. O. L. Z. V. S. Alexis Conneau, "Unsupervised Cross-lingual Representation Learning at Scale," *CoRR,* 2019.

[11] J. L. F. Y. Q. C. Y. H. W. L. X. L. Yongqiang Ma, “AI vs. Human -- Differentiation Analysis of Scientific Content Generation,” 2023.

[12] D. Fischler, “Real or fake text? We can learn to spot the difference,” March 2023. [Online]. Available: https://penntoday.upenn.edu/news/penn-seas-real-or-fake-text-we-can-learn-spot-difference.

[13] R. Reddy, “AI-Generated vs. Human-Written Text : Complete Analysis,” Ranktracker, July 2023. [Online]. Available: https://www.ranktracker.com/blog/ai-generated-vs-human-written-text-complete-analysis/.

[14] A. K, “How to Build a Machine Learning Model to Distinguish If It’s Human or ChatGPT?,” AnalyticsVidhya, May 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/04/how-to-build-a-machine-learning-model-to-distinguish-if-its-human-or-chatgpt/.

[15] D. S. H. N. J. T. R. M. T. M. D. M. F. Niful Islam, “Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning,” 2023.

OZemi at SemEval-2024 Task 1: A Simplistic Approach to Textual Relatedness Evaluation Using Transformers and Machine Translation

Hidetsune Takahashi and **Xingru Lu**
Sean Ishijima and **Deokgyu Seo** and **Yongju Kim** and **Sehoon Park** and **Min Song**
and **Kathylene Marante** and **Keitaro-Luke Iso** and **Hirotaka Tokura**
and **Emily Ohman**
Waseda University
ohman@waseda.jp

Abstract

In this system paper for SemEval-2024 Task 1 subtask A, we present our approach to evaluating the semantic relatedness of sentence pairs in nine languages. We use a mix of statistical methods combined with fine-tuned BERT transformer models for English and use the same model and machine-translated data for the other languages. This simplistic approach shows consistently reliable scores and achieves middle-of-the-pack ranks in most languages.

1 Introduction

SemEval 2024 Task 1 (Ousidhoum et al., 2024c) calls for assigning scores indicating semantic textual relatedness (STR) of sentence pairs in 14 different languages. We participate in Track A, which is the supervised subtask for systems that have been trained using the provided labeled datasets (Ousidhoum et al., 2024a). There are data in Algerian Arabic, Amharic, English, Hausa, Kinyarwanda, Marathi, Moroccan Arabic, Spanish, and Telugu for Track A and we provide a solution for all 9 languages. The labeled data has been manually annotated for relatedness using a comprehensive annotation framework (Abdalla et al., 2023).

A large portion of previous work in STR has been conducted for English-language data. This task does include English, but the focus is on lower-resourced languages (Hedderich et al., 2021; Marreddy et al., 2022). STR is a crucial component in information retrieval, summarization, and question answering, as well as in developing Large Language Models (LLMs). The lack of STR or similar NLP resources for low-resource languages means progress is often much slower in related research such as the development of LLMs too making the progress achieved through this task societally highly impactful by providing new tools and datasets for language where NLP resources are lacking (Vulić et al., 2020; Zhang et al., 2020).

Our methodology uses both traditional TF-IDF vectorization and transformer models like BERT (Devlin et al., 2018) fine-tuned for semantic relatedness tasks. We leverage the high availability of resources that exist for English to fine-tune a BERT model that we then use on machine-translated versions of the datasets for the other languages (except for Spanish where a multilingual BERT model yielded better results than with machine translating the data). This approach seems to capture both lexical patterns and deeper semantic relationships, making it effective for linguistically diverse datasets, and cost-effective because there is no need to manually annotate more than one dataset (language). It is therefore an alternative approach to creating language-specific models. Although our approach is simplistic, it has the upside of working reasonably well for any low-resource language that has some machine translation or parallel language data resources.

2 Background

In SemEval-2024 Task 1, the dataset was adapted from the STR-2022 dataset (Abdalla et al., 2023). The STR-2022 dataset contains 5,500 English sentence pairs that were manually annotated using a comparative annotation framework, yielding fine-grained scores ranging from 0 to 1 (maximally unrelated to maximally related). The dataset was constructed by sampling sentences from various sources to capture a wide range of text characteristics such as sentence structure, formality, and grammaticality. The sources include datasets on formality (Rao and Tetreault, 2018), book reviews (Wan and McAuley, 2018), paraphrases (Wieting and Gimpel, 2018), natural language inference (Bowman et al., 2015), semantic textual similarity (Cer et al., 2017), stance (Mohammad et al., 2016), and text simplification (Horn et al., 2014).

The corresponding datasets for the other languages are much smaller and consist of roughly

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 7–12

1000 sentence pairs each with minor variations in size.

Semantic **relatedness** and semantic **similarity** are closely related concepts in natural language processing (NLP), however, the terms are not interchangeable. Semantic similarity is a narrower definition that only takes term similarity into account (e.g. *fork* is similar to *knife*), whereas relatedness in addition to similarity can include terms or concepts that are related beyond hyponymic relationships such as *fork* being related to *eating*) (Asaadi et al., 2019; Batet and Sánchez, 2016). This task focuses on the broader concept of relatedness but utilizes more narrowly defined datasets based on similarity as well in the construction of the datasets.

In recent years the development of NLP resources for low-resource languages has been speeding up, but there are still large discrepancies in what types of tools, models, and resources exist for languages other than English (Hedderich et al., 2021). There are also significant differences in the resources available among low-resource languages and what being a low-resource language entails (Hämäläinen, 2021; Marreddy et al., 2022). For most of the languages in this task, there are at least some models and tools (see e.g Deode et al., 2023) but a handful of research groups working on a language is quite different from nearly all research groups in the world working on producing models and tools for a language (English). When there is a need for more data, often data augmentation methods are used to increase data points. Machine translation is an established method of data augmentation, particularly with low-resource languages where it might not be possible to use language-specific models (Amjad et al., 2020).

3 System overview

Our choice of methodology was shaped by pedagogical considerations as well as technical. As we participated in this task as part of an undergraduate senior research seminar in computational methods, we purposely started with the simplest most readily available tools progressing towards more advanced methods. Along the way, we compared the results and progress at each step in an attempt to better understand how each of the specific NLP tools worked and how accurate their output was when used on real projects such as this dataset.

The main strategy of our system is integrating classic NLP methods, such as the Dice Score and TF-IDF, with advanced deep learning techniques like BERT models, to determine semantic relatedness between sentence pairs. Firstly, our system imports a CSV dataset that contains pairs of English sentences (separated by "\n"), each paired with a relatedness score ranging from 0 to 1. Then, to assess semantic relatedness, the system adopts several basic NLP techniques, including Spacy's Linguistic Features for efficient text processing, TF-IDF for calculating word importance in sentences, Spacy Similarity and Cosine Similarity for measuring sentence similarity, and fine-tuned BERT Models for leveraging contextually rich semantic analysis (Devlin et al., 2018). These techniques collectively contribute to a robust evaluation of semantic relatedness against the given scores. We tried early on to adopt the same approach to the non-English languages with language-specific transformer-based similarity and relatedness models, but the language-specific models yielded much lower evaluation scores than what the English model achieved with machine-translated versions of the non-English datasets. We used the Google Translate API to translate the datasets into English to maintain consistency in analysis. Compared to other translation APIs such as DeepL, for this task, Google Translate seemed to produce better translations, perhaps because of how it favors more common words over context thus being more suited for STR and/or STS tasks (see e.g. Öhman, 2022).

Participating in the semantic relatedness task using the hybrid strategy allows for a comprehensive exploration of the system's performance and methodology. Through a detailed analysis, you can assess the effectiveness of traditional NLP methods, including TF-IDF and Spacy's Linguistic Features, in comparison to more advanced deep learning techniques like BERT. Evaluating the impact of contextual embeddings from fine-tuned BERT models provides insights into how well the model captures nuanced semantic relationships. The inclusion of Google Translate for non-English languages offers an opportunity to examine the system's ability to maintain consistency across languages. Assessing the generalization capability, scalability, and efficiency of the system provides a holistic understanding of its applicability to diverse datasets and real-world scenarios. Through this participation, we can uncover strengths, weaknesses, and potential areas for improvement, guiding future research directions and refining the hybrid strategy for en-

hanced semantic relatedness evaluation across languages and varied linguistic contexts. In particular, this approach shows that it is possible to achieve reasonable accuracies by leveraging the prevalence of tools and models designed for English with low-resource languages.

Our code is available on GitHub [1].

4 Experimental setup

At the beginning stage of the experiment, we undertook an examination of several readily implementable models on the English baseline dataset and compared the predicted scores with human-labeled scores through Pearson correlation scores.

In the initial English baseline model, we included the SpaCy similarity model[4], cosine vector similarity, and fine-tuned-BERT models[5]. For the SpaCy similarity, we directly applied it to the training dataset, yielding a result of 0.34 (Pearson). In the case of cosine similarity, we tried out two methods of word embedding:

1. **Binary occurrence vectors**: This approach involves creating set-based word vectors using binary occurrence, combining them into a joint space, and comparing them using cosine similarity to quantify the relatedness between the original sets in vectorized forms.

2. **TF-IDF transformer-based vectors**: Using the TF-IDF vectorizer from the sklearn (Pedregosa et al., 2011) library, we obtained TF-IDF weights for each word. The TF-IDF weight is proportional to the word's frequency in the document but is offset by its frequency in the corpus.

Upon comparing these two word-embedding methods, the Pearson correlation results did not reveal a significant difference. Therefore, we selected the Binary occurrence method as the cosine vector similarity, which achieved a score of 0.61 as indicated in Table 1. We use Pearson as opposed to Spearman rank correlation simply because that is what the original task description uses (Ousidhoum et al., 2024a,b).

The final component of the English baseline is the application of the fine-tuned BERT model to compute semantic relatedness with the (unfine-tuned) ClinicalBertSimilarity[5] and WebBertSimilarity[5] models and a batch size of 10 for both. The creators of the model claim that the "project contains an interface to fine-tuned, BERT-based semantic text similarity models. It modifies pytorch-transformers by abstracting away all the research benchmarking code for ease of real-world applicability"[5]. This proved to be the most successful approach with a result of 0.8 for English. Although the task in question is about semantic relatedness, since many of the datasets involved in the creation of the datasets come from similarity data. Additionally, as similarity can be considered a subtype of relatedness, the use of similarity models seemed logical due to their wider availability compared to relatedness models.

After establishing the English baseline, we evaluated several multilingual and language-specific BERT-based similarity models to assess textual relatedness (or similarity) across other language training datasets including the SBERT model for Telugu (Joshi et al., 2022), Sentence-BERT (Reimers and Gurevych, 2019), BioLORD-2023(Remy et al., 2023), etc. However, the results were suboptimal, which is surprising since previous work has shown that sentence transformers show significant improvements to semantic similarity tasks, particularly cross-lingual tasks (Hämmerl et al., 2023). Given the significantly better performance of the English baseline, we decided to translate all language datasets into English before applying the relatedness prediction models. In the case of Spanish we found that using distiluse-base-multilingual-cased-v1(Reimers and Gurevych, 2019) produced higher accuracies than the translation approach, and thus Spanish is the only language we did not translate to English.

When introducing the translation tools, we explored two approaches: utilizing a translation model (Machine Translation) and implementing Google Translate.

1. **Machine Translation**: In the Machine Translation method, we applied M2M100 (Fan et al., 2020) as the translation model. The model can directly translate between the 9,900 directions of 100 languages.

2. **Google Translate**: For the machine translations, we utilized the deep-translator library[6],

[1] https://github.com/esohman/SemEval2024

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2 and Reimers and Gurevych (2020)

[4] https://spacy.io/usage/linguistic-features#vectors-similarity

[5] https://github.com/AndriyMulyar/semantic-text-similarity

	Train Data					Dev Data	
LANGUAGE	English Translation			Multilingual Model		Official score	Ranking
	Spacy Similarity	cos vector	fine-tuned SBERT	DBMCv1[2]	all-mpnet-base-v2[3]		
Algerian Arabic	0.25	0.44	0.51	0.42	0.39	0.37	18/20
Amharic	0.37	0.61	0.78	0.16	0.12	0.78	11/16
English	*0.34*	*0.61*	*0.80*	*	*	*0.81*	*10/34*
Hausa	0.07	0.43	0.65	0.21	0.34	0.62	12/19
Kinyarwanda	0.18	0.39	0.57	0.3	0.38	0.57	8/14
Marathi	0.45	0.68	0.81	*	*	0.86	13/25
Moroccan Arabic	-0.01	0.45	0.34	0.34	0.16	0.45	18/19
Spanish	0.58	0.7	0.66	*	*	0.62	8/17
Telugu	0.44	0.67	0.78	0.36	0.29	0.78	16/24

Table 1: Task scores for different methods

a versatile tool that facilitates simple language translation using multiple translators.

Despite the relatively high performance claimed by the M2M100 model as described by Fan et al. (2020), the results after the translation process are less than 0.5 for all languages except Spanish, where it achieved a result of 0.67. In contrast, the Google Translate API demonstrated better performance during the training process with the English baseline model (detailed results are listed in Table 1).

Our multilayered approach mirrors that of Jeyaraj and Kasthurirathna (2021) although ours is a much simpler setup.

5 Results

Our rankings show that our approach is nowhere near the state-of-the-art, but it is still a reliable option when more language-specific approaches are unavailable as is often the case with moderately low-resource languages. Out team ranked in the middle of the pack for most languages, but in the top third for English, Marathi, and Spanish, and the bottom for both Arabic dialects, which was expected. The rankings, scores, and models used for each submission can be seen in table 1. We analyze the results in the conclusions section.

6 Conclusions

To sum up, we first focused on English to have a good solution with fine-tuned BERT, and then we applied that solution to other languages by translating the sentences into English using machine translation. Since our English solution is reasonably good (rank 10/34, official score of .81), the application of the solution worked much better than using multilingual models in many languages including Amharic, Marathi, and Telugu for which there exist language-specific semantic similarity models. We speculate that the reason the *MT+English model* worked better than the language-specific relatedness models is due to the higher quality and more diverse training data for the English model(s) as well as machine translation simplifying words to the most commonly used ones, artificially making similar sentences more similar.

The importance of an accurate machine translation can be seen in the failure of our approach with the Arabic dialects in particular. Google Translate does not have specific translators for Moroccan or Algerian Arabic, instead, we had to rely on general Arabic. This likely produced much lower quality translations obfuscating the semantic links between the sentence pairs making it difficult for the English model to accurately judge relatedness. This issue was further exacerbated by the fact that no one on our team speaks any of the languages in the task besides English, which made manual evaluations of the MT output difficult.

Darja and Darija are the names for Algerian and Moroccan Arabic respectively, and they are collectively known as Maghrebi Arabic. Due to its roots in Berber languages, there are notable distinctions between Maghrebi Arabic and Standard Arabic, and using the latter for these two dialects may yield a suboptimal result.

Curiously, a similar issue occurred with Spanish. Spanish is much more closely related to English than the other languages in subtask A, and therefore we expected our approach to get a fairly high score similar to English, especially considering the current state of machine translation between English

[6] https://deep-translator.readthedocs.io/en/latest/README.html#id1

and Spanish. However, it seems that translation of Spanish into English affects the semantic relations of the original sentences, which might be one of the main reasons causing the very low scores and making us choose the multilingual model for Spanish rather than the machine-translated one.

We hypothesize that one of the reasons that Google Translate worked so well on the low-resource languages most dissimilar from English might be because smaller training datasets for MT would force the translation to use less context and instead increase the reliance on individual lexical items leading to sentence pairs with high relatedness becoming more similar via translation. For languages with better MT models, it is conceivable that the better translations work against this approach as it might make the sentence pairs less similar as reflected by the higher scores for Spanish using multilingual models, and the very low scores for both Arabic dialects. In future work, it might be worthwhile to use mixed methods starting with language-specific models and then expanding to incorporate machine translation and larger models developed for, e.g., English.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Maaz Amjad, Grigori Sidorov, and Alisa Zhila. 2020. Data augmentation using machine translation for fake news detection in the urdu language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2537–2542.

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. 2019. Big bird: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 505–516.

Montserrat Batet and David Sánchez. 2016. Improving semantic relatedness assessments: Ontologies meet textual corpora. *Procedia Computer Science*, 96:365–374. A paper we can use to show that we understand the difference between semantic relatedness and semantic similarity and looked into potential approaches/methods in order to improve the accuracy of our work.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023. L3cube-indicsbert: A simple approach for learning cross-lingual sentence representations using multilingual bert. *arXiv preprint arXiv:2304.11434*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation.

Mika Hämäläinen. 2021. Endangered languages are not low-resourced! *arXiv preprint arXiv:2103.09567*.

Katharina Hämmerl, Alina Fastowski, Jindřich Libovickỳ, and Alexander Fraser. 2023. Exploring anisotropy and outliers in multilingual language models for cross-lingual semantic sentence similarity. *arXiv preprint arXiv:2306.00458*.

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463.

Manuela Nayantara Jeyaraj and Dharshana Kasthurirathna. 2021. Mnet-sim: A multi-layered semantic similarity network to evaluate sentence similarity. *arXiv preprint arXiv:2111.05412*.

Ananya Joshi, Aditi Kajale, Janhavi Gadre, Samruddhi Deode, and Raviraj Joshi. 2022. L3cube-mahasbert and hindsbert: Sentence bert models and benchmarking bert sentence representations for hindi and marathi. *arXiv preprint arXiv:2211.11187*.

Mounika Marreddy, Subba Reddy Oota, Lakshmi Sireesha Vakada, Venkata Charan Chinni, and Radhika Mamidi. 2022. Am i a resource-poor language? data sets, embeddings, models and analysis for four different nlp tasks in telugu language. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(1):1–34.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. A dataset for detecting stance in tweets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3945–3952.

Emily Öhman. 2022. Self & feil: Emotion lexicons for finnish. In *Proceedings of the 6th Digital Humanities in the Nordic and Baltic Countries conference. CEUR Workshop Proceedings*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024c. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

François Remy, Kris Demuynck, and Thomas Demeester. 2023. BioLORD: Semantic textual representations fusing llm and clinical knowledge graph insights.

Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. 2021. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. *arXiv preprint arXiv:2010.05731*.

Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 86–94, New York, NY, USA. Association for Computing Machinery.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.

L3i++ at SemEval-2024 Task 8: Can Fine-tuned Large Language Model Detect Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text?

Hanh Thi Hong Tran[1,2,3] and **Tien Nam Nguyen**[1]
and **Antoine Doucet**[1] and **Senja Pollak**[2]
[1] University of La Rochelle, L3i, La Rochelle, France
[2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
{firstname.lastname}@univ-lr.fr, senja.pollak@ijs.si

Abstract

This paper summarizes the participation of the L3i laboratory of La Rochelle University (L3i++) in *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection*. In this task, we aim to solve two over three Subtasks: (1) Monolingual and Multilingual Binary Human-Written vs. Machine-Generated Text Classification; and (2) Multi-Way Machine-Generated Text Classification. We propose a comparative study among three groups of methods to trigger the detection: (1) Using metric-based models; (2) Using a fine-tuned sequence-labeling language model (LM); and (3) Using a fine-tuned large-scale language model (LLM). Our findings show that LLM surpassed the performance of traditional sequence-labeling LM as the benchmark and metric-based approaches. We ranked $5^{th}/62$ in Multilingual Binary Human-Written vs. Machine-Generated Text Classification and $6^{th}/70$ Multi-Way Machine-Generated Text Classification on the leaderboard. Our code is publicly available at https://github.com/honghanhh/semeval8.

1 Introduction

The rise of large language models (LLMs) has led to a significant step forward in producing remarkably controllable, fluent, and grammatical text, triggering a surge in machine-generated content across diverse platforms such as news, social media, question-answering forums, educational, and even academic contexts. Notably, recent LLMs like ChatGPT[1] and GPT-4 (OpenAI, 2023) exhibit a remarkable ability to generate coherent and contextually appropriate responses to a wide array of user queries.

Unfortunately, use and abuse come hand in hand. Although the fluency of these generated texts positions LLMs as potential candidates for replacing human labor in numerous applications, this has also raised concerns about their potential for misuse, particularly in spreading misinformation and causing disruptions within the education system. Given that humans struggle to distinguish between machine-generated and human-written text, it becomes imperative to develop automated systems capable of identifying machine-generated text to curb the risks associated with its misuse.

In this paper, as the participants in *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection* (Wang et al., 2024), we investigate the feasibility of training a classifier that can reliably differentiate between text generated by humans and text that appears human-like but is generated by machines in two paradigms:

- Subtask A: Given a full text, determine whether it is human-written or machine-generated in monolingual (only English sources) and multilingual versions.
- Subtask B: Given a full text, determine who generated it (human-written or generated by a specific language model).

To address these problems, we explore the performance of diverse methodologies, which can be divided into three categories, including:

- Five different metric-based methods: Log-Likelihood, Rank, Log-Rank, Entropy, and DetectGPT (He et al., 2023).
- Two traditional sequence-labeling language models: monolingual RoBERTa$_{large}$[2] (Liu et al., 2019) and multilingual XLM-R$_{large}$[3] (Conneau et al., 2020).
- A large language model (LLM): $LLaMA-2-7b-hf$[4] (LLaMA-2) (Touvron et al., 2023).

[1] https://chat.openai.com/

[2] FacebookAI/roberta-large

[3] FacebookAI/xlm-roberta-large

[4] NousResearch/Llama-2-7b-hf

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 13–21

This paper is organized as follows. We present related work in Section 2, followed by Section 3, where we introduce the data used to solve this challenge. Our proposed methods are described in Section 4 before we present our findings and an error analysis in Section 5. Finally, in Section 6 we present our conclusions, and future work and discuss the limitations of the proposed methods.

2 Related Work

The success of LLMs in various downstream NLP tasks (Perez et al., 2021; Vilar et al., 2022; Hegselmann et al., 2023) leads to the overuse and abuse of the information generated by LLMs. However, it is essential to acknowledge that the outputs generated by LLMs are not always accurate, giving rise to the issue of hallucination (Azamfirei et al., 2023). Consequently, there is a need for clear differentiation in addressing this concern.

To address these issues, researchers have developed several automatic detection methods (Badaskar et al., 2008; Zellers et al., 2019; Ippolito et al., 2020; Uchendu et al., 2021) that can identify the machine-generated text from the human-written text, which initially can be divided into two categories, i.e., metric-based methods and model-based methods.

2.1 Metric-based methods

Metric-based methods leverage pre-trained LLMs to process the text and extract distinguishable features from it, e.g., the rank or entropy of each word in a text conditioned on the previous context. Then, predicted distribution entropy determines whether a text belongs to machine-generated or human-written texts. Some metric-based detection methods include Log-Likelihood, Rank, Entropy, GLTR, Log-Rank, and DetectGPT (He et al., 2023), to cite a few.

2.2 Model-based methods

In the model-based methods (Zellers et al., 2019; Habibzadeh, 2023; Guo et al., 2023), the classification models are trained using a corpus that contains both machine-generated or human-written texts to make predictions, for example, ChatGPT Detector (Guo et al., 2023), GPTZero (Habibzadeh, 2023), LM Detector (Ippolito et al., 2020), to mention a few.

Regarding *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection* (Wang et al., 2024), RoBERTa (Liu et al., 2019) and XLM-R (Conneau et al., 2020) are two language models that can be considered as the baseline for these specific tasks.

2.3 Challenges

Yet, there is currently no existing framework capable of automatically distinguishing between human-written and machine-generated texts at both binary and multi-way paradigms outlined in the described tasks as well as no existing free available architecture taking advantage of recent open-sourced LLMs to tackle the issue.

3 Data

We work on two datasets provided by *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection* (Wang et al., 2024), whose statistics covering the number of examples for each source and each label are presented in Tables 1 and 2 for Subtask A and B, respectively.

Labels	**Human**				**Machine**			
Source	**Monolingual**		**Multilingual**		**Monolingual**		**Multilingual**	
	Train	**Dev**	**Train**	**Dev**	**Train**	**Dev**	**Train**	**Dev**
arxiv	15,498	500	15,998	-	11,999	500	14,999	-
peerread	2,357	500	2,857	-	9,374	500	11,708	-
reddit	15,500	500	16,000	-	12,000	500	14,999	-
wikihow	15,499	500	15,999	-	12,000	500	15,000	-
Wikipedia	14,497	500	14,997	-	11,033	500	14,032	-
bulgarian	-	-	6,000	-	-	-	6,000	-
chinese	-	-	6,000	-	-	-	5,934	-
urdu	-	-	3,000	-	-	-	2,899	-
indonesian	-	-	2,995	-	-	-	3,000	-
russian	-	-	-	1,000	-	-	-	1,000
arabic	-	-	-	500	-	-	-	500
german	-	-	-	500	-	-	-	500
Total	63,351	2,500	83,846	2,000	56,406	2,500	88,571	2,000

Table 1: Subtask A

In Subtask A of the monolingual version, both the training and development sets are sourced from the same data group for both labels. However, in the multilingual version of Subtask A and Subtask B, the development set is sourced from different places compared to the training set.

For both versions of Subtask A, data were collected from diverse sources, leading to label imbalances. For example, in the monolingual Subtask A training set, there is a notable scarcity of samples from *peerread* compared to the other sources. Conversely, in Subtask B, the dataset is balanced.

Labels	Source	Train	Dev	Labels	Source	Train	Dev
Human	arxiv	2,998	-	**davinci**	arxiv	2,999	-
	reddit	3,000	-		reddit	2,999	-
	wikihow	2,999	-		wikihow	3,000	-
	Wikipedia	3,000	-		Wikipedia	3,000	-
	peerread	-	500		peerread	-	500
total		11,997	500				
chatGPT	arxiv	3,000	-	**bloomz**	arxiv	3,000	-
	reddit	3,000	-		reddit	2,999	-
	wikihow	3,000	-		wikihow	3,000	-
	Wikipedia	2,995	-		Wikipedia	2,999	-
	peerread	-	500		peerread	-	500
total		11,995	500	*total*		11,998	500
cohere	arxiv	3,000	-	**dolly**	arxiv	3,000	-
	reddit	3,000	-		reddit	3,000	-
	wikihow	3,000	-		wikihow	3,000	-
	Wikipedia	2,336	-		Wikipedia	2,702	-
	peerread	-	500		peerread	-	500
total		11,336	500	*total*		11,702	500

Table 2: Subtask B

4 Methodology

This section tackles the problem by formulating it as supervised classification tasks. We then introduce our proposed solution architecture for each task, covering the models used, and present how we fine-tuned them with hyperparameter configurations, and how we assessed their performance.

4.1 Problem Statements

4.1.1 Subtask A

We formulate the problem at hand as a binary supervised classification task, whose objective is to learn a mapping between a representation of the text and a binary variable, which is 1 if the text is machine-generated, and 0 otherwise. Mathematically, we learn a function f that, given an input text t_i, represented as a set of features $[f_1^i, ..., f_k^i]$, outputs an estimated label $\hat{l}_i \in \{0, 1\}$, i.e., $\hat{l}_i = f(t_i)$. Note that Subtask A covers two versions: monolingual and multilingual versions.

4.1.2 Subtask B

Similarly, we consider the task as a supervised classification where we aim to learn a function f that, given an input text t_i, represented as a set of features $[f_1^i, ..., f_k^i]$, outputs an estimated label $\hat{l}_i \in \{0, 1, 2, 3, 4, 5\}$, i.e., $\hat{l}_i = f(t_i)$ where 0 refers to the human-written texts and the rests are those generated by different machines, including 1-*ChatGPT*, 2-*cohere*, 3-*davinci*, 4-*bloomz*, and 5-*dolly*, respectively.

Furthermore, we are interested in gaining insights from the classifier's predictions that allow us to understand which features contribute positively to detecting machine-generated text.

4.2 Our architecture

The overall architecture of our proposed approach is demonstrated in Figure 1. The general idea is to use a machine learning model trained to discriminate between text samples generated by a human and text samples generated by LLMs. Different directions could be pursued to extract useful features from a text and perform text classification.

4.2.1 Metric-based models

Inspired the works from He et al. (2023) and Spiegel and Macko (2023), we capture the local information from the texts using the following methods: (1) *Log-Likelihood*, (2) *Rank*, (3) *Log-Rank*, (4) *Entropy*, and (5) *MFDMetric*.

- *Log-Likelihood*: Given a text, we average the token-wise log probability of each word generated from a language model to generate a score for this text.
- *Rank*: For each word in a text, given its previous context, we calculate the absolute rank of this word. Then, for a given text, we compute the score of the text by averaging the rank value of each word.
- *Log-Rank*: Slightly different from the Rank metric that uses the absolute rank, the Log-Rank score is calculated by first applying the log function to the rank value of each word.
- *Entropy*: Similar to the Rank score, the Entropy score of a text is calculated by averaging the entropy value of each word conditioned with its previous context.
- *Multi-Feature Detection Metric* or *MFDMetric*: This is a two-step zero-shot method that (1) considers four distributional information (*Log-Likelihood*, *Log-Rank*, *Entropy*), and statistical information (*LLM-Deviation*) as input features; and (2) classify the text using neural networks.

In *Log-Likelihood*, a larger score denotes the text is more likely to be machine-generated. Meanwhile, in *Rank* and *Log-Rank*, a smaller score denotes the text is more likely to be machine-generated. Similarly, the machine-generated text is more likely to have a lower *Entropy* score. Note that metric-based methods are only applied to Subtask A.

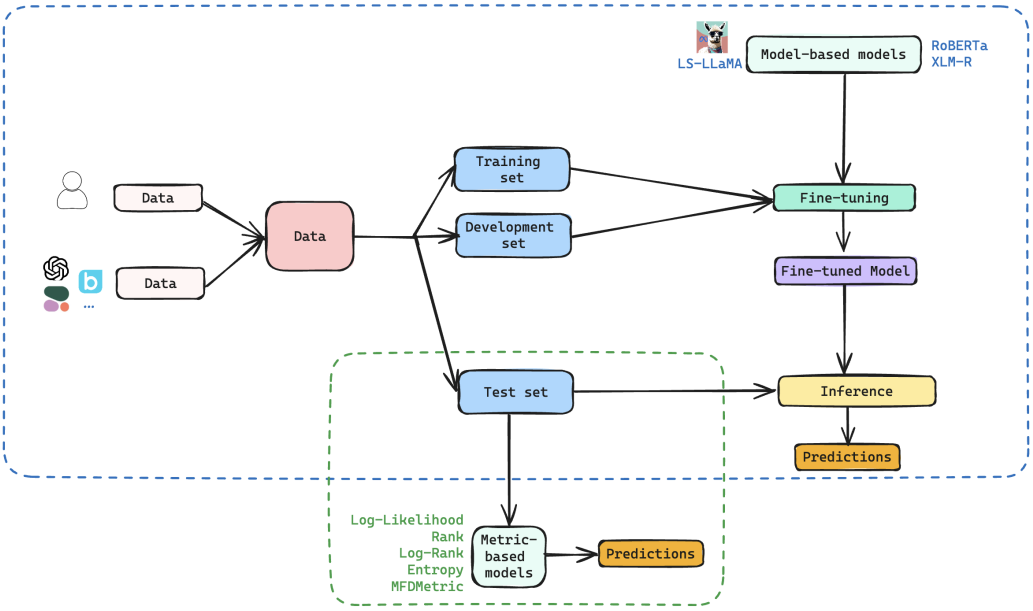


Figure 1: Our general architecture for Subtask A (both blue and green boxes) and Subtask B (only blue box).

4.2.2 Model-based models

LMs Two Transformer-based models have been fine-tuned as sequence classifiers, namely RoBERTa (Liu et al., 2019) and XLM-R (Conneau et al., 2020). RoBERTa is a Transformers model pretrained on a large corpus of English data in a self-supervised fashion using a masked language modeling (MLM) objective. Meanwhile, XLM-R is a multilingual version of RoBERTa that was pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages. These models are also suggested as the baseline methods from *SemEval-2024 Task 8* organizers.

LLMs Given the recent success of the LLMs architectures for solving downstream NLP tasks, we decided to follow the same vein to build our classifier. As such, we start with LLaMA-2 (Touvron et al., 2023), an LLM model pre-trained for the sequence classification task, using its corresponding tokenizer to preprocess data. We then fine-tune the model on the training subset of collected data. Consequently, the fine-tuned model is used for inference on the testing subset. Finally, the obtained classification scores are evaluated against the ground truth.

4.3 Hyperparameters

Metric-based models We took advantage of IMGTB[5] framework with default parameter settings suggested from He et al. (2023) and Spiegel and Macko (2023).

LMs We fine-tuned 2 LMs, namely RoBERTa and XLM-R, using HuggingFace Transformers Pytorch Trainer with the following configuration: batch size = 16, learning rate = 1e-5, weight decay = 0.01, number of epoch = 10.

LLaMA-2 To make the comparison comparable, we fine-tuned LS-LLaMA[6] (version: *LLaMA-2-7b-hf*) using the HuggingFace Transformers PyTorch Trainer class with the same configuration: batch size = 16, learning rate = 1e-5, and the number of epochs = 10 with max length = 256 and Lora = 12.

All the experiments were implemented on an NVIDIA RTX A6000 with CUDA Version of 12.0 and 49140MiB.

4.4 Evaluation metrics

For both Subtasks, we use *Accuracy*, *macro-F1*, and *micro-F1* as the evaluation metrics to measure our classifiers' performance. These are also the standard metrics in *SemEval-2024 Task 8*, which makes our works more comparable with other participants. We assess the performance of the development sets first and apply the best models to the test set. The final leaderboard reported results only for *Accuracy*.

[5] https://github.com/michalspiegel/IMGTB

[6] https://github.com/4AI/LS-LLaMA

Methods	Subtask A - Mono			Subtask A - Multi			Subtask B		
	Accuracy	Micro F1	Macro F1	Accuracy	Micro F1	Macro F1	Accuracy	Micro F1	Macro F1
Metric-based methods									
Log-Likelihood	0.51880	0.40011	0.51880	0.49700	0.46172	0.49700	-	-	-
Rank	0.71760	0.71760	0.71262	0.51000	0.51000	0.47705	-	-	-
Log-Rank	0.51700	0.38751	0.51700	0.49675	0.49675	0.46197	-	-	-
Entropy	0.53880	0.43979	0.53880	0.49475	0.45385	0.49475	-	-	-
MFDMetric	0.65820	0.63645	0.65820	0.49450	0.45875	0.4945	-	-	-
Language model (LM)-based methods - Benchmarks from competition									
RoBERTa	0.65920	0.65920	0.61629	0.49100	0.49100	0.48721	0.73167	0.73167	0.69539
XLM-R	0.75740	0.75740	0.75130	0.52275	0.52275	0.48949	0.60267	0.60267	0.56838
Large language model (LLM)-based methods									
LS-LLaMA$_{2-7b-hf}$	**0.81500**	**0.81500**	**0.80862**	**0.87400**	**0.87400**	**0.87399**	**0.75500**	**0.75500**	**0.73165**

Table 3: Performance of Subtask A (monolingual and multilingual versions) and Subtask B on development set where the training set is split into training and validation set with the ratio of 8:2 for training progress.

5 Results and Discussion

Table 3 demonstrates the evaluation of different methods on the development set before the test set was released, while Table 4 reports our final performance on the test set in comparison with the baseline suggested by *SemEval-2024 Task 8* and our approach ranking on the leaderboard.

Methods	A - Mono	A - Multi	B
Baseline	**0.88466**	0.80887	0.74605
LS-LLaMA$_{2-7b-hf}$	0.85840	**0.92867**	**0.83117**
Our ranking	**25**/125	**5**/62	**6**/70

Table 4: Our performance in *Accuracy* on the test set with the same train-validation-test split of *SemEval Task8*.

5.1 General Observations

We first present different experiment results on the development set in Table 3. We observed that overall, LLM-based methods, such as LS-LLaMA$_{2-7b-hf}$, tend to outperform other approaches across all sequence classification tasks, suggesting the effectiveness of leveraging large pre-trained language models for these tasks. Meanwhile, metric-based methods have varying performance, with *Rank* showing some competitiveness, but generally, they are outperformed by LLM and LM-based methods. Regarding LM-based approaches, XLM-R tends to surpass the performance of RoBERTa in the monolingual version of Subtask A despite RoBERTa being specifically designed for English only.

Based on the performance of the development set, we applied LS-LLaMA$_{2-7b-hf}$, which yields superior performance in these Subtasks compared to other methods, to the test set. As shown in Table 4, despite not surpassing the baseline of Subtask A's monolingual version, our models significantly outperform the baseline of Subtask A's multilingual version and Subtask B with approximately 10% gain on average. While we ranked only 25^{st} over 125 participants in the monolingual version of Subtask A, we demonstrate competitive performance to be ranked 5^{th} over 62 and 6^{th} over 70 participants in the multilingual version of Subtask A and Subtask B, respectively.

We conducted several analyses to investigate how different factors would affect the detection performance of our best classifier.

5.2 Effect of Text Length

We first present the distribution of the number of words (*#. words*) for predicted human-generated and machine-generated texts (*Predictions*) and their ground truth (*GT*) in the dataset in each Subtask (shown in Figure 2).

On ground-truth levels, Figure 2 highlights discrepancies in word distribution between human-written texts and those generated by different LLMs. This is evident in Subtask A by the difference in word count distribution between human and machine-generated labels and in Subtask B by the varying generated performance of individual LLMs compared to human-written ones. For instance, *davinci* can generate long-context answers (more than 2500 words) while others respond in more concise ways (less than 1500 words).

Despite these discrepancies, compared predictions against ground truth, our classifier effectively captures the distribution of generated texts per

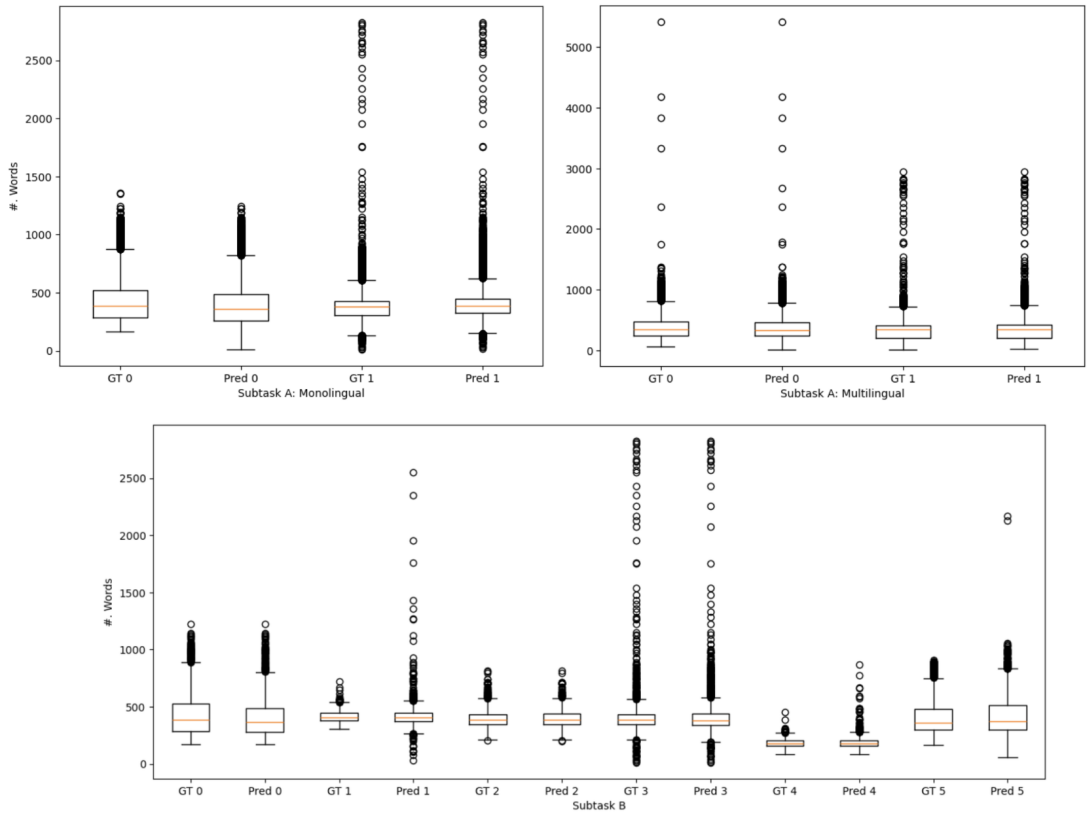


Figure 2: The distribution of words (*#. words*) for human-written and machine-generated texts of our predictions (*Pred*) and the ground truth (*GT*) on different datasets of different tasks (Subtask A: 0-*Human*, 1-*Machine*; Subtask B: 0-*Human*, 1-*ChatGPT*, 2-*cohere*, 3-*davinci*, 4-*bloomz*, and 5-*dolly*).

class, resulting in comparable word distributions between predictions and ground truth except in *ChatGPT* and *dolly* where most of the examples we misclassified are outliers.

5.3 Class-wise Performance

To better investigate the detection performance of different classes, we visualize the normalized confusion matrix of different tasks when we used our LLaMA-2 classifier as shown in Figure 3.

On one hand, in terms of Accuracy, unlike the multilingual version of Subtask A where all the classes can be well detected with up to 94% in Accuracy, the monolingual version suffers significantly from misclassifying human-written texts into machine-generated ones, which reduces the performance of the overall classifier (the accuracy of the human-written class falls into around 76%). Most of the misclassified texts are human-written that our classifier mistakenly took for the machine-generated ones.

On the other hand, when it comes to multi-way machine-generated text classification as Subtask B, the predictive performance of our classifier varies depending on the type of LLMs used to generate texts. Although LLaMA-2 has a good performance in identifying human-written and machine-generated texts generated by *ChatGPT*, *bloomz*, and *dolly*, the performance in attributing machine-generated texts from other LLMs (e.g., *cohere*, and *davinci*) is largely limited. For example, the prediction accuracy of *ChatGPT*, *bloomz* is almost perfect (99.53% and 99.70%, respectively). Meanwhile, that of *cohere* is just above the average (around 60%) and its texts are often misclassified as machine-generated texts from *davinci*, followed by *ChatGPT*. This is expected due to potential overlap in the distribution of the metric among various LLMs, which introduces extra challenges in attribution.

Broadly speaking, our findings suggest that the fine-tuned LLMs (e.g., LLaMA-2) excel in detecting machine-generated multilingual texts and accurately classifying machine-generated texts within a specific category, (e.g., *ChatGPT*, *bloomz*, *dolly*). However, they do exhibit challenges in detecting

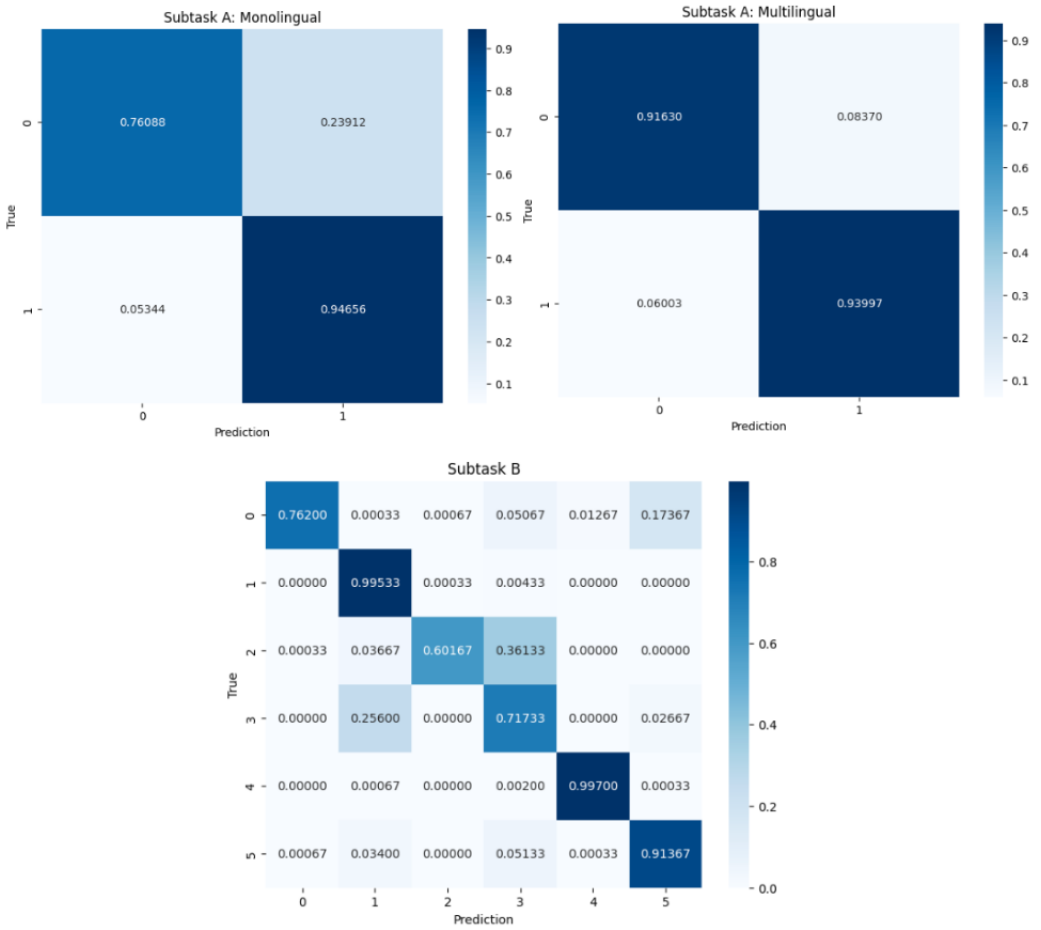


Figure 3: Normalized confusion matrix of LLaMA-2 methods on different tasks. Note that the values in the diagonal represent the class-wise accuracy (Subtask A: 0-*Human*, 1-*Machine*; Subtask B: 0-*Human*, 1-*ChatGPT*, 2-*cohere*, 3-*davinci*, 4-*bloomz*, and 5-*dolly*).

them in other categories (e.g., *cohere*, and *davinci*). Further studies are needed to improve the lower-performing classes.

6 Conclusions

In conclusion, this paper outlines our contribution to the first two Subtasks of *SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection*, namely Monolingual and Multilingual Binary Human-Written vs. Machine-Generated Text Classification and Multi-Way Machine-Generated Text Classification. We conducted a comprehensive comparative study across three methodological groups: Five metric-based models (Log-Likelihood, Rank, Log-Rank, Entropy, and MFD-Metric), two fine-tuned sequence-labeling language models (RoBERTA and XLM-R); and a fine-tuned large-scale language model (LS-LLaMA).

Our findings suggest that our LLM outperformed both traditional sequence-labeling LM benchmarks and metric-based approaches. Furthermore, our fine-tuned classifier excelled in detecting machine-generated multilingual texts and accurately classifying machine-generated texts within a specific category, (e.g., *ChatGPT*, *bloomz*, *dolly*). However, they do exhibit challenges in detecting them in other categories (e.g., *cohere*, and *davinci*). This is due to potential overlap in the distribution of the metric among various LLMs. Overall, we ranked 6^{th} in both Multilingual Binary Human-Written vs. Machine-Generated Text Classification and Multi-Way Machine-Generated Text Classification on the leaderboard.

In future work, we would like to take a step further to evaluate whether our classifier is robust enough against adversarial attacks (e.g., paraphrasing, random spacing, adversarial perturbation) as

well as investigate how to make our model more interpretable and explainable, which is important, but insufficiently addressed when detecting machine-generated contents.

Limitations

Regarding specificity and domain dependence, our classifier might not effectively distinguish among different types of machine-generated texts, such as texts generated by different models, for different purposes, or in specific domains (which can be seen in the case of detecting texts generated by *cohere* and *davinci*).

Acknowledgements

The work was partially supported by the Slovenian Research and Innovation Agency (ARIS) core research program Knowledge Technologies (P2-0103) and projects Linguistic Accessibility of Social Assistance Rights in Slovenia (J5-50169) and Embeddings-based techniques for Media Monitoring Applications (L2-50070). The work has also been supported by the ANNA (2019-1R40226) and TERMITRAD (2020-2019-8510010) projects funded by the Nouvelle-Aquitaine Region, France. Besides, the work was supported by the project Cross-lingual and Cross-domain methods for Terminology Extraction and Alignment, a bilateral project funded by the program PROTEUS under the grant number BI-FR/23-24-PROTEUS006.

References

Razvan Azamfirei, Sapna R Kudchadkar, and James Fackler. 2023. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):1–2.

Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Farrokh Habibzadeh. 2023. Gptzero performance in identifying artificial intelligence-generated medical texts: A preliminary study. *Journal of Korean Medical Science*, 38(38).

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

OpenAI. 2023. Gpt-4 technical report.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

Michal Spiegel and Dominik Macko. 2023. Imgtb: A framework for machine-generated text detection benchmarking. *arXiv preprint arXiv:2311.12574*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2022. Prompting palm for translation: Assessing strategies and performance. *arXiv preprint arXiv:2211.09102*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti,

Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

nicolay-r at SemEval-2024 Task 3: Using Flan-T5 for Reasoning Emotion Cause in Conversations with Chain-of-Thought on Emotion States

Nicolay Rusnachenko
Newcastle Upon Tyne, England
rusnicolay@gmail.com

Huizhi Liang
Newcastle University
Newcastle Upon Tyne, England
huizhi.liang@ncl.ac.uk

Abstract

Emotion expression is one of the essential traits of conversations. It may be self-related or caused by another speaker. The variety of reasons may serve as a source of the further emotion causes: conversation history, speaker's emotional state, etc. Inspired by the most recent advances in Chain-of-Thought, in this work, we exploit the existing three-hop reasoning approach (THOR) to perform large language model instruction-tuning for answering: emotion states ($\text{THOR}_{\text{STATE}}$), and emotion caused by one speaker to the other ($\text{THOR}_{\text{CAUSE}}$). We equip $\text{THOR}_{\text{CAUSE}}$ with the reasoning revision (RR) for devising a reasoning path in fine-tuning. In particular, we rely on the annotated speaker emotion states to revise reasoning path. Our final submission, based on $\text{Flan-T5}_{\text{base}}$ (250M) and the rule-based span correction technique, preliminary tuned with $\text{THOR}_{\text{STATE}}$ and fine-tuned with $\text{THOR}_{\text{CAUSE-RR}}$ on competition training data, results in 3rd and 4th places ($F1_{\text{proportional}}$) and 5th place ($F1_{\text{strict}}$) among 15 participating teams. Our THOR implementation fork is publicly available: https://github.com/nicolay-r/THOR-ECAC

1 Task Overview

Extracting potential causes that lead to emotion expressions in text is the crucial aim of Emotion Cause Extraction (ECE) domain (Xia and Ding, 2019). In particular, the SemEval-2024 Task 3 (Wang et al., 2024) is aimed at emotion-cause pair analysis in conversations from the sitcom *Friends*. The conversations are organized into Emotion-Cause-in-Friends dataset (Wang et al., 2023) and includes the JSON-formatted training ($\text{TRAIN}_{\text{json}}$) and evaluation ($\text{TEST}_{\text{json}}$) parts. The authors propose 6 emotion classes to annotate: (i) speaker emotion states, and (ii) emotion caused by one utterance to the other. These classes are: $E = \{\text{SURPRISE, SADNESS, JOY, DISGUST, FEAR, ANGER}\}$, and NEUTRAL for absence of emotion. We denote $E' = E \cup \{\text{NEUTRAL}\}$ as a complete set.

Among the several subtasks of ECAC-2024, in this paper we focused on *Subtask 1*: textual emotion-cause pair extraction in conversations. In this subtask, each conversation represents a list of utterances. Every utterance (u) yields the following: utterance text (u_{text}), speaker name ($u_{speaker}$), emotion state ($u_{state} \in E'$), and ID (u_{id}). The annotation of the emotion cause pairs represents a list $P = [p_1 \dots p_{|P|}]$, in which each pair $p \in P$ is a labeled source-target[1] tuple $p = \langle u^{src}, u^{tgt}, e_c \rangle$, where $e_c \in E$.

We initiate our studies by analyzing the training data ($\text{TRAIN}_{\text{json}}$) for the subject of annotated emotion-cause pairs $\langle u^{src}, u^{tgt} \rangle$ in it, and report:

1. Quantitative statistics of the mentioned emotion-cause pairs (Table 1);
2. Distance statistics (in utterances) between u^{src} and u^{tgt} (Table 2);
3. Distribution statistics between speaker state (u_{state}) and emotion *speaker causes* ($e^{u \to *}$) (Table 3).

According to the Table 2, most emotion was found to be caused by such utterances u^{src} that are the same as or mentioned before u^{tgt} ($\delta \geq 0$). Therefore, given $\langle u^{src}, u^{tgt} \rangle$ we denote its context $X = \{u^1 \dots u^k\}$ as a *history* of the past $k-1$ utterances of u^{tgt}, where $u^{tgt} = u^k \in X$, $u^{src} \in X$.
Task definition: Given an emotion-causing utterance pair within context $\langle u^{src}, u^{tgt}, X \rangle$ answer the emotion $e_c \in E'$ caused by u^{src} towards u^{tgt}.

2 Methodology

We propose a two-stage training mechanism for performing instruction-tuning on large language models (LLMs), aimed at accurately inferring of

[1] Spans-prediction is beyond the scope of our methodology.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 22–27

Parameter	Value
Conversations (total)	1374
Emotion causes pairs per conversation	6.46
Emotion causes pairs in annotation (total)	8879
Self-cause per conversation (% from total)	**51.86%**
Self-cause by different utterance (% from total)	12.83%

Table 1: Quantitative statistics of the emotion-cause pairs in the competition training data (TRAIN$_{\text{json}}$)

Parameter	future	past				
$\delta = u_{id}^{tgt} - u_{id}^{src}$	< 0	0	1	2	3	4
Causes count	377	4605	2759	810	332	160
Average per δ	0.12	3.35	2.01	0.59	0.24	0.12
Covering (%)	–	51.9	82.9	92.1	**95.8**	97.6

Table 2: Distance statistics (δ) (in utterances) between source (u^{src}) and target (u^{tgt}) of emotion-cause pairs in the competition training data (TRAIN$_{\text{json}}$)

$u_{state} \backslash e^{u \to *}$	JOY	SUR	ANG	SAD	DIS	FEA
total	2653	2092	1984	1336	518	296
JOY	.89	.06	.03	.01	.01	.00
SURPRISE	.07	.78	.07	.03	.03	.02
ANGER	.01	.07	.83	.06	.02	.02
SADNESS	.02	.09	.06	.81	.01	.01
DISGUST	.03	.07	.14	.06	.70	.01
FEAR	.02	.13	.08	.05	.04	.68
NEUTRAL	.24	.38	.22	.08	.04	.03

Table 3: . Distribution statistics between speaker state (u_{state}) and emotion *speaker causes* ($e^{u \to *}$) in the competition training data (TRAIN$_{\text{json}}$); values in each row are normalized

the task answers. Given triplet $\langle u^{src}, u^{tgt}, X \rangle$ of emotion-cause pair $\langle u^{src}, u^{tgt} \rangle$ in context X, the proposed mechanism aims at LLM instruction-tuning, in order to answer $e \in E'$ that refers to:

STAGE 1: emotion state u_{state}^{tgt};

STAGE 2: emotion cause by u^{src} to u^{tgt}.

Therefore, for emotion-cause pairs extraction we use the STAGE 2 towards the model tuned in STAGE 1 to infer $e_c \in E'$ caused by u^{src} towards u^{tgt}.

Instead of directly asking LLM the final result at each stage, we exploit the Chain-of-Thought (CoT) concept in the form of the Three-hop Reasoning (THOR) framework (Hao et al., 2023). We believe that LLM can infer the span that conveys emotion and opinion about it before answering $e \in E'$. Figure 1 illustrates the proposed training methodology, empowered by the CoT prompting. We refer to the instruction-tuning mechanisms of the STAGE 1 and STAGE 2 as THOR$_{\text{STATE}}$ and THOR$_{\text{CAUSE}}$ respectively.

2.1 Chain-of-Thought Prompting

We adopt the THOR framework (Hao et al., 2023) in LLM fine-tuning with the prompt templates adapted for emotion-cause pair analysis in conversations. We define the intermediate *span* (s) and latent *opinion* expression (o). With $C_i, i \in \overline{1..3}$ we denote the prompts that wrap the content in the input context. The construction of stages is as follows.

THOR$_{\text{STATE}}$ This is a STAGE 1 of the proposed training methodology, aimed at preliminary LLM instruction-tuning. Given $\langle u^{tgt}, X \rangle$, we apply the following three steps to infer $u_{state}^{tgt} = e'_1 \in E'$:

Step 1: s'_1 = [$C_1(X)$, which text spans are possibly causes emotion on u_{text}^{tgt}?]

Step 2: o'_1 = [$C_2(C_1, s'_1)$. Based on the common sense, what is the implicit opinion towards the mentioned text spans that causes emotion on u_{text}^{tgt}, and why?]

Step 3: e'_1 = [$C_3(C_2, o'_1)$. Based on such opinion, what is the emotion state of u_{text}^{tgt}?]

where s'_1 could be interpret as $s'_1 = argmax\ p(s_1|X, u_{text}^{tgt})$, latent opinion o'_1 as $o'_1 = argmax\ p(o_1|X, u_{text}^{tgt}, s'_1)$, and the final answer e'_1 noted as: $e'_1 = argmax\ p(e_1|X, u_{text}^{tgt}, s'_1, o'_1)$.

THOR$_{\text{CAUSE}}$ This is a STAGE 2 of the proposed methodology, based on emotions-cause pairs. We use this stage for (i) *fine-tuning* and (ii) task result *inferring* purposes. Given context $\langle u^{src}, u^{tgt}, X \rangle$ we omit[2] $u^{tgt} \in X$ from the input parameters by referring to it as «*end of the conversation*». We apply the following steps to infer $e'_2 \in E'$ caused by u^{src} to u^{tgt}:

Step 1: s'_2 = [$C_1(X)$, which specific text span of u_{text}^{src} is possibly causes emotion?]

Step 2: o'_2 = [$C_2(C_1, s'_2)$. Based on the common sense, what is the implicit opinion towards the cause of mentioned text span of u_{text}^{src}, and why?]

Step 3: e'_2 = [$C_3(C_2, o'_2)$. Based on such opinion, what is the emotion caused by source towards the last conversation utterance?]

[2]To reduce the problem statement to the one for which THOR was originally designed (Pontiki et al., 2016)

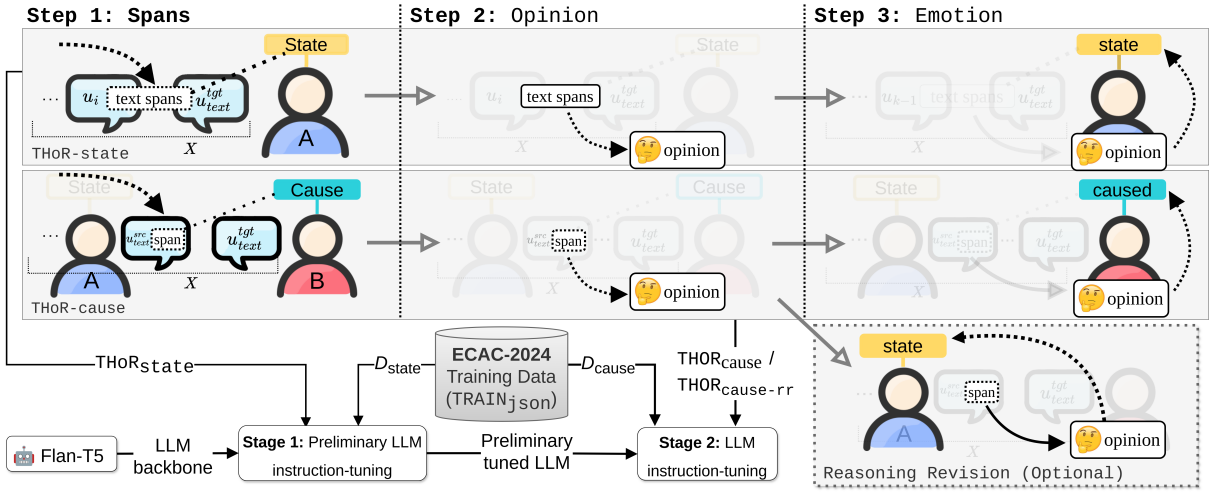


Figure 1: Two-stage LLM tuning methodology for inferring emotion caused by u^{src} towards u^{tgt} in context X by adapting THoR (Hao et al., 2023) to reason and answer: (i) u^{tgt}_{state} (THoR$_{\text{STATE}}$), and (ii) emotion caused by u^{src} towards u^{tgt} (THoR$_{\text{CAUSE}}$), optionally enhanced by Reasoning-Revision and by predicting u^{src}_{state} (THoR$_{\text{CAUSE-RR}}$).

where s'_2 could be interpret as $s'_2 = argmax\ p(s'_2|X, u^{src}_{text})$, opinion o' could be interpret as $o'_2 = argmax\ p(o_2|X, u^{src}_{text}, s'_2)$, and the final answer e'_2 noted as: $e'_2 = argmax\ p(e_2|X, u^{src}_{text}, s'_2, o'_2)$.

2.2 Reasoning Revision with Supervision

During the LLM instruction-tuning process with the THoR, it is possible to devise a reasoning path. Technically, at each step of the chain we have all the necessary information to query our model with the final answer. With the following approach, we believe in a better model alignment on state-cause dependency (Table 3): speakers are likely to cause an emotion, similar to their states[3]. To revise this knowledge, in this paper, we impute the following prompt to support our opinion O, obtained at the end of the THoR$_{\text{CAUSE}}$ step 2 (Fig. 1):

Step 3.1: u'^{src}_{state} =[$C_3(C_2, o'_2)$, Based on such opinion, what is the emotion state of u^{src}_{text}?]

Due to the definition of the task, we believe in the correctness of this knowledge within the emotion cause task. Once step 3.1 is embedded, the result answer $e'_2 \in E'$ in THoR$_{\text{CAUSE}}$ from the step 3 could be reinterpret as $e'_2 = argmax\ p(e_2|X, u^{src}_{text}, s'_2, o'_2, u'^{src}_{state})$. We refer to this setup as THoR$_{\text{CAUSE-RR}}$.

[3]Except NEUTRAL speaker state (Table 3)

3 Datasets and Experiential Setup

We adopt textual resources provided by the competition organizers (Wang et al., 2024): training (TRAIN$_{\text{json}}$) and evaluation (TEST$_{\text{json}}$) data. Within TRAIN$_{\text{json}}$, for each conversation, we rely on (i) speakers *emotion states*, and (ii) *emotion causes* annotation to compose the datasets D_{state} and D_{cause}, respectively. Each dataset represent a list of tuples $t = (u, X, L)$, where u is an utterance of the conversation context $X = \{u^1 \dots u^k\}$, and L is a list of emotion labels, defined as:

- $L = [u^k_{state}]$ in the case of D_{state} ($u^k_{state} \in E'$)
- $L = [u_{state}, e^u]$ in the case of D_{cause}, where e^u is emotion expressed by u towards u^k, or NEUTRAL otherwise ($e^u \in E'$)

D_{state} represent entries of all possible utterances in all conversations with their emotional states $u_{state} \in E$. For the particular utterance u, we consider its context as $X_u = \{u' : u_{id} - u'_{id} \leq k\}$.

D_{cause} includes all possible pairs $\langle u^{src}, u^{tgt} \rangle$, where $u^{src}_{id} \leq u^{tgt}_{id}$, and $u^{tgt}_{id} - u^{src}_{id} \leq k$. For the particular pair, we compose the related context (X') as follows: $X' = \{u' : u^{tgt}_{id} - u'_{id} \leq k\}$. For each pair, we assign $e \in E$ if the pair is present in conversation annotation and NEUTRAL otherwise. We rely on the analysis in Table 2 to limit the number of pairs, as well as the size of the context. We set $k = 3$ to cover 95.8% emotion-cause pairs. We also cover the case of emotions caused from within the same utterance (59.5%, see Table 1). As for emotions caused by the same speaker of

Source	TRAIN$_{\text{json}}$		TEST$_{\text{json}}$
Part	train	dev	test
D_{state} (total)	12144	1475	
NEUTRAL	5299	630	.
JOY	2047	254	.
SURPRISE	1656	184	.
ANGER	1423	192	.
SADNESS	1011	136	.
DISGUST	372	42	.
FEAR	336	37	.
D_{cause} (total)	30445	3612	15794
NEUTRAL	23750	2765	15794
JOY	2111	279	–
SURPRISE	1725	202	–
ANGER	1307	174	–
SADNESS	932	120	–
DISGUST	387	47	–
FEAR	233	25	–

Table 4: Statistics of the composed datasets D_{state} and D_{cause} from the publicly available competition data, for the two training methodology stages respectively; statistics is listed for $k = 3$.

other utterance, we assess that excluding this type of pairs (12.83%, according to Table 1), results in $\approx 23\%$ pairs reduction of D_{cause} and hence reduces training time. Therefore, the result D_{cause} excludes pairs of this type in train, dev and test parts.

Table 4 lists the statistics of the composed resources. We use the 9:1 proportion for TRAIN$_{\text{json}}$ to compose train and dev, respectively. To represent $X \in t$, we concatenate its representation of utterances. For each utterance $u \in X$, we use the following formatting template: «$u_{speaker}$: u_{text}». To represent utterance $u \in t$, we refer to u_{text}. For each $l \in L$ formatting, we utilize its lowercase text value. The implementation details for the datasets preparation are publicly available.[4]

Setup. We follow the publicly available framework setups (Hao et al., 2023) and adopt encoder-decoder style instructive Flan-T5[5] as our backbone LLM for the proposed methodology. We experiment with a 250M (base) version. For evaluations on dev, we adopt the F1-measure for E', denoted as $F1(E')$. The evaluation on test assessed with the set of $F1$-metrics, provided by the competition organizers (details in Section 4). We consider the instruction-tuning of the Flan-T5 model with the following techniques: conventional PROMPT, THOR (Section 2.1), and THOR$_{\text{CAUSE}}$ with reasoning revision (Section 2.2). To conduct the experiment, we rent a server with a single NVIDIA A100 GPU (40GB). We set temperature 1.0, learning rate $2 \cdot 10^{-4}$, optimizer AdamW (Loshchilov and Hutter, 2017), BATCH-SIZE of 32.

For the PROMPT technique, we use the template «$C_1(X)$. $I(u)$. Choose from E'», where $I(u)$ corresponds to the instruction. For D_{cause} we use $I(u)$ = «What emotion causes u_{text} towards the last conversation utterance?»

[4] https://github.com/nicolay-r/SemEval2024-Task3

[5] https://huggingface.co/google/flan-t5-base

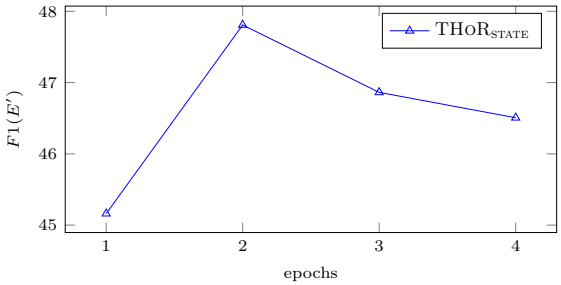


Figure 2: Result analysis of the preliminary fine-tuning of Flan-T5$_{\text{base}}$ on D_{state} dev using THOR$_{\text{STATE}}$ technique per epoch by $F1(E')$

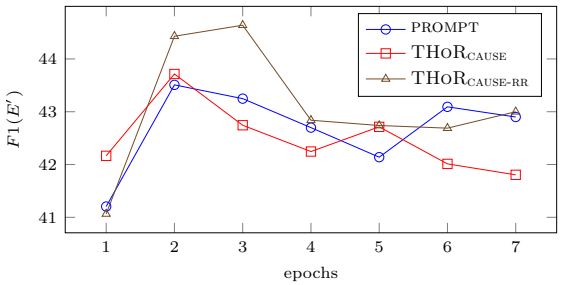


Figure 3: Flan-T5$_{\text{base}}$† fine-tuning results comparison by $F1(E')$ on D_{cause} dev part per each epoch across fine-tuning techniques: PROMPT, THOR$_{\text{CAUSE}}$, and THOR$_{\text{CAUSE-RR}}$.

4 Experiments

Stage 1. Figure 2 illustrates the analysis of the $F1$ on dev part during the preliminary tuning of Flan-T5$_{\text{base}}$ on D_{state}.[6] We investigate the overfitting after 2 epochs of training. The best state, obtained at the end of the epoch #2 with the $F1(E')$= 47.81 on the D_{state}-dev part, has been selected. In further, we refer to this model as Flan-T5$_{\text{base}}$†.

Stage 2 Figure 3 provides a comparative analysis of different fine-tuning techniques. As at the pre-training stage, we investigate the ability to learn task emotion states 2-3 training epoch, followed by overfitting. Switching from PROMPT to THOR$_{\text{CAUSE-RR}}$ technique, we investigate the improvement by 2.5% percent by $F1(E')$ on the dev

[6] We left the comparison with other pre-training techniques listed in 3 out of scope of this paper due to alignment with the CoT concept in STAGE 2.

Source	dev		test		
Model	$F1(E')$	$F1^w_s$	$F1^w_p$	$F1_s$	$F1_p$
PROMPT					
FT5$_{\text{base}}$‡	43.51	9.68	22.27	10.05	22.21
THOR$_{\text{CAUSE}}$					
FT5$_{\text{base}}$‡	43.72	–	–	–	–
THOR$_{\text{CAUSE-RR}}$					
FT5$_{\text{base}}$‡	44.64	9.74	23.54	10.33	23.94
THOR$_{\text{CAUSE-RR}}$ + *Algorithm-based Spans Correction*					
FT5$_{\text{base}}$‡	44.64	12.86	24.28	13.26	24.13

Table 5: Evaluation results for Flan-T5$_{\text{base}}$‡ on dev and test parts of the D_{cause} dataset; the results of the final submission are highlighted in gray

part of the D_{state} dataset. We refer to the best fine-tuned versions as Flan-T5$_{\text{base}}$‡, separately per each fine-tuning technique in Table 5 (dev column).

The official evaluation includes the following $F1$ measures: (i) weighted averaged $F1^w_*$ / non-weighted ($F1_*$), and (ii) strict ($F1_s$) / not-strict ($F1_p$) towards predicted spans. To form the submissions for official evaluation, the following span corrections approaches were used: (i) punctuation terms[7] exclusion from utterance prefixes and suffixes (by default), and (ii) algorithm-based (Section 4.1). Table 5 (test columns) illustrate the available results of T5$_{\text{base}}$‡ in official evaluation. **Final submission** represents the results of Flan-T5$_{\text{base}}$‡ (THOR$_{\text{CAUSE-RR}}$ technique), and application of algorithm-based spans correction.

4.1 Algorithm-based Spans Correction

Our methodology (Section 2) is limited on utterance level emotion cause prediction.[8] We believe it is reflected in the relatively low results of $F1_s$ on the test dataset (see Table 5). Therefore, we analyze TRAIN$_{\text{json}}$ and adopt a placeholder solution, aimed at enhancing the results by $F1_s$.

We apply a *rule-based approach* based on differences between the original utterance texts and their span annotations in the training data. Using TRAIN$_{\text{json}}$, we compose *prefix-* (V_p) and *suffix-* (V_s) vocabularies. For vocabulary entries, we select those that satisfy all of the following criteria: (i) the length of entry does not exceed 5 words, (ii) entry starts (in the case of V_s), or ends (in the case of V_p) with the punctuation sign[7].

[7]We use string.punctuation preset in Python

[8]Technically it is possible to obtain spans (Section 2), however we could not investigate the practical valuty of the THOR$_{\text{CAUSE}}$-based Flan-T5$_{\text{base}}$‡ responses from step #1.

Parameter	Value
Conversations (total)	2917
Emotion causes pairs in annotation	665
Average per conversation	4.39

Table 6: Quantitative statistics of the automatically extracted emotion-cause pairs by Flan-T5$_{\text{base}}$‡ (THOR$_{\text{CAUSE-RR}}$ technique) from the evaluation data (TEST$_{\text{json}}$)

Parameter	past			
$\delta = u^{tgt}_{id} - u^{src}_{id}$	0	1	2	3
Causes count	1711	1012	148	46
Average per δ	2.57	1.52	0.22	0.07
Covering (%)	58.7	93.3	98.4	100.0

Table 7: Statistic of distances in utterances (δ) between source (u^{src}) and target (u^{tgt}) of emotion-cause pairs for automatically extracted emotion-cause pairs by Flan-T5$_{\text{base}}$‡ (THOR$_{\text{CAUSE-RR}}$ technique) from the evaluation data (TEST$_{\text{json}}$)

$u_{state} \backslash e^{u \to *}$	JOY	SUR	ANG	SAD	DIS	FEA
JOY	.87	.08	.02	.01	.01	.00
SURPRISE	.09	.75	.06	.05	.03	.01
ANGER	.05	.14	.68	.08	.03	.01
SADNESS	.06	.11	.03	.76	.02	.02
DISGUST	.07	.11	.07	.05	.68	.01
FEAR	.00	.15	.09	.02	.00	.74
NEUTRAL	.36	.40	.07	.12	.03	.02

Table 8: Distribution statistics between speaker state (u_{state}) and emotion *speaker causes* ($e^{u \to *}$) for automatically extracted emotion-cause pairs by Flan-T5$_{\text{base}}$‡ (THOR$_{\text{CAUSE-RR}}$ technique) from the evaluation data (TEST$_{\text{json}}$); values in each row are normalized

$u_{state} \backslash e^{* \to u}$	JOY	SUR	ANG	SAD	DIS	FEA
JOY	.97	.01	.01	.00	.01	.00
SURPRISE	.04	.89	.04	.01	.01	.01
ANGER	.04	.05	.83	.05	.02	.01
SADNESS	.02	.02	.03	.89	.02	.01
DISGUST	.02	.04	.05	.07	.81	.01
FEAR	.00	.06	.07	.04	.03	.80
NEUTRAL	.60	.13	.03	.16	.05	.02

Table 9: Distribution statistics between speaker state (u_{state}) and emotion *caused on them* ($e^{* \to u}$), for automatically extracted emotion-cause pairs by Flan-T5$_{\text{base}}$‡ (THOR$_{\text{CAUSE-RR}}$ technique) from the evaluation data (TEST$_{\text{json}}$); values in each row are normalized

For each utterance text (u_{text}) that causes emotion, we compose an updated u'_{text} by applying: (1) correction of u_{text} prefixes with V_p, followed by (2) correction of suffixes from V_s for the results from (1). We alter u'_{text} in the case of $u'_{text} = \emptyset$. The algorithm 1 illustrates an implementation for the prefixes correction with V_p.[9]

[9]Implementation is publicly available in https://github.com/nicolay-r/SemEval2024-Task3

Algorithm 1 Emotion-cause prefixes correction for u_{text}

```
updated ← True
V_p' ← sorted V_p by decreased entry lengths in words
while u_text ≠ ∅ or updated do
    updated ← False
    u'_text ← u_text                    ▷ Modified version of u_text
    for v_p ∈ V_p' do
        if u_text ends with v_p then
            u'_text ← part of u_text before v_p
            updated ← True
            break
        end if
    end for
end while
```

4.2 Final Submission Analysis

We report the following emotion-cause pairs $\langle u^{src}, u^{tgt} \rangle$ analysis results for the Flan-T5$_{\text{base}}$‡ (THOR$_{\text{CAUSE-RR}}$ technique, final submission):

1. Quantitative statistics of the extracted emotion-cause pairs (Table 6);
2. Distance statistics (in utterances) between u^{src} and u^{tgt} (Table 7);
3. Distribution statistics between speaker state (u_{state}) and the emotion *speaker causes* ($e^{u \to *}$) (Table 8);
4. Distribution statistics between speaker state (u_{state}) and emotion *caused on them* ($e^{* \to u}$) (Table 9).

According to the results in Table 8, we observe that the correlation between the state of the speaker u utterance (u_{state}) and the emotion it causes ($e^{u \to *}$) **is similar to** the related statistics on the competition training data (Table 3). We also investigate the alignment of the speaker states (u_{state}) with the emotion caused on them ($e^{* \to u}$) and the precision of the result varies between 80-97% (Table 9). The known source of misalignment is the case when emotion[10] $e^{* \to u} \in E$ caused on u with $u_{state} = \text{NEUTRAL}$ (bottom row, Table 9).

5 Conclusion

In this paper, we present a Chain-of-Thought (CoT) methodology aimed at fine-tuning LLM for emotion state and cause extraction. We consider the problem of *emotion cause analysis in conversations* as a context-based problem with the mentioned utterance that causes emotion towards the last utterance in context. We devise our CoT for emotion causes and propose a reasoning revision methodology aimed at imputing the speaker emotion to support the decision on caused emotion. Our CoT represent a Three-hop Reasoning approach priory known as THoR. We apply this approach to fine-tune LLM and predict: (i) emotion state of the mentioned utterance, and (ii) emotion caused by mentioned utterance towards the last utterance in context. We experiment with the Flan-T5$_{\text{base}}$ (250M) model fine-tuning using resources provided by task organizers. The application of CoT with reasoning revision allows us to improve the results by 2.5% (F1-measure) compared to prompt-based tuning. In further work, we expect to contribute with the: (i) analysis of larger models, and (ii) enhanced reasoning revision techniques, mentioned in the final submission analysis.

[10] JOY especially, as the most frequently appearing class.

References

Fei Hao, Li Bobo, Liu Qian, Bing Lidong, Li Fei, and Chua Tat-Seng. 2023. Reasoning implicit sentiment with chain-of-thought prompting. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1171–1182.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammed AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)*, pages 19–30. Association for Computational Linguistics.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

StFX-NLP at SemEval-2024 *Task* 9: BRAINTEASER: Three Unsupervised Riddle-Solvers

Ethan Heavey, James Hughes, Milton King
St. Francis Xavier University
{eheavey, jhughes, mking}@stfx.ca

Abstract

In this paper, we explore three unsupervised learning models that we applied to *Task* 9: BRAINTEASER of SemEval 2024. Two of these models incorporate word sense disambiguation and part-of-speech tagging, specifically leveraging SensEmBERT and the Stanford log-linear part-of-speech tagger. Our third model relies on a more traditional language modelling approach. The best performing model, a bag-of-words model leveraging word sense disambiguation and part-of-speech tagging, secured the 10^{th} spot out of 11 places on both the sentence puzzle and word puzzle subtasks.

1 Introduction

Riddles often exploit the commonsense of the solver to lead them astray, subverting expectations with it's answer. For example, the riddle "A young girl fell off of a 20 foot ladder but wasn't hurt. How? *She fell off of the bottom rung.*" leads the solver astray by including the height of the ladder in the initial question, tricking one into latching onto misleading information. *Task* 9: BRAINTEASER (Jiang et al., 2024) presents riddles to a predictive model and asks the model to choose one of four answers to the riddle, in the hopes of bridging the gap between vertical and lateral thinking (Waks, 1997) within language models. The data provided for the *Task* is written in English and was obtained from public websites by utilizing web crawlers (Jiang et al., 2023).

The three models we employ to solve this task all apply an unsupervised learning approach, with two of the three models leveraging word senses and part-of-speech tagging to aid in their predictive capabilities. We wanted to leverage the senses of the nouns in the question and in each possible answer as we hypothesized that the senses present in the question and each answer may aid our models in piercing the proverbial commonsense veil that makes brainteasers and riddles difficult to begin with.

Our best approach, the bag-of-words model, landed us in 10^{th} place out of 11 places in the "overall" results of both subtasks. While 13 teams competed, two teams tied for both 2^{nd} and 4^{th} place in the sentence subtask, two teams also tied for both 1^{st} and 11^{th} place in the word subtask results.

Our code can be found on Github[1].

2 Background

BRAINTEASER places emphasis on the ability of a predictive model to use vertical and lateral thinking. Vertical thinking leverages logic and rationality to perform a sequential analysis of a problem, whereas lateral thinking (or "thinking outside the box") leverages creativity to solve problems. The *Task* is divided into two subtasks — sentence puzzles and word puzzles. We applied our models to both, with each subtask requiring vertical and lateral thinking to solve. Figure 1 breaks down how sentence and word puzzles can be solved with lateral thinking. The train of thought labeled with a red "X" demonstrates logical thinking based on the information available at the time, whereas the alternate thought process — the line of thinking that allows the solution to be derived — displays how lateral thinking can affect the answer to a riddle as more context is provided.

The dataset associated with the *Task* presents each sample as a question and four possible answers. Table 1 shows an example of both a sentence puzzle question and its possible answers, and a word puzzle question and its possible answers. Each sample also has two variants; a semantic reconstruction and a context reconstruction. These reconstructions are designed to further test a model's reasoning ability.

[1] https://github.com/VeiledTee/BrainTeaser

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 28–33

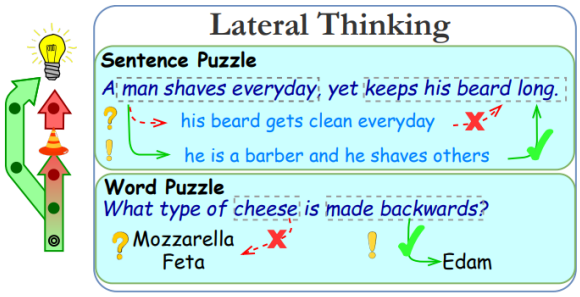


Figure 1: An example of how lateral thinking can be used to solve sentence and word puzzles. Figure taken from BRAINTEASER system paper (Jiang et al., 2023).

Whilst the training and development sets contain extra information regarding the correct answer, our unsupervised approaches only required the test set. Not using the labeled training and validation data, while limiting our models, allows them to be more versatile in situations where labeled data is not available.

Word Sense Disambiguation (WSD) is a natural language processing (NLP) task that involves determining the correct meaning or sense of a word within a given context (Navigli, 2009). Many words in natural language have multiple senses, and WSD aims to identify the intended sense of a word in a specific sentence or context. This is used in various language processing applications, such as machine translation, information retrieval, and text summarization. We employ WSD by leveraging SensEmBERT (Scarlini et al., 2020), coupled with WordNet (Fellbaum, 1998) to disambiguate the sense of a token in a particular context.

SensEmBERT is a knowledge-based approach to WSD that produces high-quality sense embeddings. WordNet is a large lexical database that organizes words and their meanings into sets of interlinked synonyms called synsets.

We leverage part-of-speech (POS) tagging in order to determine which tokens in each question and answer are nouns we can determine the sense of. We employ the English version of the Stanford Log-Linear POS Tagger[2] (Toutanova et al., 2003) — which leverages dependency networks to aid in tagging tokens — in this work. For the purposes of our work, we only work with nouns — tokens whose tag begins with "NN".

[2] https://nlp.stanford.edu/software/tagger.shtml

3 System Overview

The following is a description of each approach we took in an attempt to solve the *Task*. We implemented a bag-of-words, language modelling, and a sense comparison approach. The language model at the core of all three of our approaches is `bert-large-cased` (Devlin et al., 2018), the same model leveraged by Scarlini et al. (2020) in the creation of SensEmBERT.

3.1 Bag of Words with WSD Approach

Our bag-of-words (BOW-WSD) model combines POS tagging with WSD to create a bag of words for the question and each possible answer. When presented with a question (q), the model creates a list containing the most prevalent sense for each noun in the question — q_senses — by leveraging Algorithm 1. Note; in this algorithm, it is necessary to concatenate the embedding of each noun to itself in order to match the format of the WordNet senses, allowing said WordNet senses to be compared to and leveraged. From q_senses, we create q_bag by removing all stop and duplicate words. Token order and context is preserved during the generation of q_senses but not for the creation of q_bag.

The process used to create q_bag is then repeated four times — once for each possible answer — creating five bags of words in total, one q_bag and an $answer\_bag$ for each of the four answers. Each $answer\_bag$ is compared to q_bag through an overlap calculation — the number of common tokens across both bags — shown in Equation 1. For example, if q_bag is "[hair, shave, beard, cut, trade]" and one of the $answer_bag$s is "[trade, cut, hair, someone]", the overlap score would be 0.667 — three overlapping tokens of nine possible tokens. The $answer\_bag$ with the highest overlap score is predicted to be the correct answer.

$$\text{avg\_overlap} = \frac{2 \cdot (|\text{bag1} \cap \text{bag2}|)}{(|\text{bag1}| + |\text{bag2}|)} \quad (1)$$

3.2 Language Modelling Approach

In the example shown in Table 1, the correct answer can be read as a natural continuation of the question — contrary to the other possible answers which do not make logical sense if appended onto the end of the question. We explore this intuition with our language modelling approach, which takes each answer, concatenates it to the end of the question, and calculate the probability of the text from

Question	Choices
Sentence Puzzle Example	
A man shaves everyday, yet keeps his beard long.	*He is a barber.* He wants to maintain his appearance. He wants his girlfriend to buy him a razor. None of the above.
Word Puzzle Example	
What part of London is in France?	*The letter N.* The letter O. The letter L. None of the above.

Table 1: An example of a sentence puzzle and a word puzzle from the BRAINTEASER dataset. The correct answer for each puzzle is in italics.

Algorithm 1: WordNet Sense Extraction

Input: Input sentence
Output: WordNet senses of nouns in the sentence
`bert-large-cased` tokenizes input
Perform POS tagging on tokenized input
$filtered\_nouns \leftarrow$ nouns from the POS tagging results
$final\_senses \leftarrow []$
for n *in* $filtered\_nouns$ **do**
    Concatenate the noun's token embedding to itself /* This format matches that of WordNet, permitting querying */
    Search WordNet for the most similar sense key using cosine similarity
    Use sense key to retrieve WordNet sense of n
    Append n_sense to $final\_senses$
Return $final\_senses$

each answer following the question using BERT (`bert-large-cased`)[3]. The predicted answer is the one associated with the largest probability.

3.3 Sense Comparison Approach

In this approach we leverage an unsupervised WSD model that makes predictions by comparing the senses of nouns. Once the primary sense of each noun in the question is identified, we utilize the `bert-large-cased` model to retrieve the embedding of the `[CLS]` token for each identified sense. This procedure is replicated for every potential answer, and the cosine similarity is employed to compute a similarity score for each pairing of `[CLS]` tokens between the senses of the question and those of each individual answer. Subsequently, these similarity scores are aggregated and averaged based on the number of senses being assessed in the current computations, both for the question and the answer. The predicted answer is the one with the highest average similarity score. Algorithm 2 outlines the steps this approach takes in further detail.

Beyond the data provided by the *Task* organizers, we leveraged the English stop words available through the NLTK Python library[4] (Bird et al., 2009), and the senses provided by WordNet[5] (Fellbaum, 1998).

4 Experimental Setup

As previously mentioned, we only use the test set in our experiments. Due to the unsupervised nature of

[3] https://huggingface.co/bert-large-cased

[4] https://www.nltk.org/

[5] https://wordnet.princeton.edu/

Algorithm 2: Sense Comparison

Input: question, list of four possible answers
Output: Predicted answer
$q\_senses \leftarrow$ WORDNETSENSEEXTRACTION(*question*)
$q\_CLS \leftarrow$ [embedding for $sense$ in q_senses] `// calculated by bert-large-cased`
$answers \leftarrow [choice_1, choice_2, choice_3, choice_4]$
$answer\_similarity \leftarrow []$
for a *in* $answers$ **do**
    $a\_senses \leftarrow$ WORDNETSENSEEXTRACTION(a)
    $a\_CLS \leftarrow$ [embedding for $sense$ in a_senses] `// calculated by bert-large-cased`
    $total\_similarity \leftarrow 0$;
    for $q\_CLS\_embedding$ *in* q_CLS **do**
        for $a\_CLS\_embedding$ *in* a_CLS **do**
            $similarity\_score \leftarrow$ COS_SIM$(q_CLS_embedding, a_CLS_embedding)$;
            $total\_similarity \leftarrow total\_similarity + similarity\_score$;
    $answer\_similarity[i] \leftarrow \frac{total\_similarity}{len(q\_CLS) \cdot len(a\_CLS)}$
$max\_index \leftarrow$ index of max element in answer_similarity
Return answers[max_index]

our approaches, the labels are not required to train our models as none of them had hyperparameters to tune.

4.1 Libraries used

Table 3 shows the Python libraries and their versions used for this *Task*. Python version 3.10.11 was used. The full `requirements.txt` file is available in our GitHub repository[6] for the project.

4.2 Evaluation Measures

The *Task* uses six metrics for both the sentence and word puzzles — 12 total — of metrics to evaluate a model's ability to solve brainteasers. The three different types of questions (original, semantic reconstruction, context reconstruction) were evaluated individually and in two groups. For a model to predict a sample in one of the groups (original and semantic reconstruction, original and semantic reconstruction and context reconstruction) correctly, all of the samples in said group must be predicted correctly.

5 Results

The performances of our models, the provided baseline models, and the best performing models submitted to this *Task* are found in Table 2.

Our BOW-WSD model (Section 3.1), the best performing of our three approaches, was able to surpass the RoBERTa-L baseline in 2 of 6 of the sentence puzzle categories, and outperforms the same baseline on 5 of 6 of the word puzzle categories. BOW-WSD outperforms or comes very close to outperforming the RoBERTa-L baseline in both "Overall" categories. The performance of our unsupervised models didn't approach the ChatGPT or Human baselines in any category. The closest our models got to the ChatGPT baseline was in the original word puzzle category with a difference of 0.155, whereas the closest our models got to the Human baseline was in the context sentence puzzle category with a difference of 0.469. The numbers achieved by our BOW-WSD model netted us 10^{th} place overall in the sentence puzzle subtask.

We suspect the relationship between the tokens in the question senses and the tokens in the correct answer's senses allowed our BOW-WSD model to outperform our other approaches. Using the sentence puzzle in Table 1 as an example, the WordNet sense of the noun "barber" (available below) from the correct answer has two tokens that overlap with the question, leading to this answer achieving a higher score than other nouns that don't overlap.

> a hairdresser who cuts hair and shaves beards as a trade

Our language modelling approach outperformed the RoBERTa-L baseline in 4 of 6 of the word puzzle categories, but did not perform well in any of

[6] https://github.com/VeiledTee/BrainTeaser

	Sentence Puzzle						Word Puzzle					
Test set	**Original**	**Semantic**	**Context**	**Orig. + Sem.**	**Orig. + Sem. + Con.**	**Overall**	**Original**	**Semantic**	**Context**	**Orig. + Sem.**	**Orig. + Sem. + Con.**	**Overall**
Best overall	1.00	.975	.925	.975	.900	.967	.969	.938	1.00	.938	.938	.969
Human	.907	.907	944	.907	.889	.920	.917	.917	.917	.917	.900	.917
ChatGPT	.608	.593	.679	.507	.397	.627	.561	.524	.518	.439	.292	.535
RoBERTa-L	.435	.402	.464	.330	.201	.434	.195	.195	.232	.146	.061	.207
BoW	.425	.400	.475	.350	.200	.433	.406	.219	.344	.125	.063	.323
LM	.225	.200	.375	.075	.050	.267	.438	.250	.500	.125	.031	.396
SC	.175	.200	.350	.175	.125	.242	.156	.063	.219	.063	.031	.146

Table 2: The accuracy scores achieved by our models (Bag-of-Words, Language Model, and Sense Comparison) on each sub-category of the test dataset. Approaches in gray are shown for comparison: the best scoring participant model for each individual category; the participant model that performed best in both the sentence and word puzzle subtasks; and the organizer's ChatGPT, RoBERTa-L, and Human baselines.

the sentence puzzle categories. We suspect that the way the word puzzles are structured lends more to the language modelling approach than the sentence puzzle structure as all the word puzzles in the test set are structured as questions — adding each answer to the end of the question can provide the language modelling approach with enough context to choose the correct answer. We believe the more succinct nature of the word puzzle problems allowed our language modelling technique to outperform our BOW-WSD model on 4 of 6 word puzzle categories, netting us 10^{th} place in the word puzzle subtask too.

Our sense comparison model unfortunately performed worse than all our models and the *Task* organizers' baselines. Our idea to leverage the senses of nouns in the sentences did not perform well when applied to this *Task*.

6 Conclusion

Whilst the best of our unsupervised models surpassed only one of the established baselines, we have been able to show that word sense disambiguation may have a place in riddle-solving models. Our BOW-WSD model performed better on the sentence puzzles, but our language modelling approach performed better on the word puzzle subtask. The inherent logical reasoning large language models obtain through the copious amount of training data they're trained on can be led astray by the information provided by a riddle. Leveraging word sense disambiguation we attempt to isolate the meaning of each noun and compare and contrast said meanings to those present in each possible answer.

In the future, we will explore other means of incorporating WSD models within our riddle-answering model along with an ensemble method. While our unsupervised approaches didn't perform well compared to other submitted models on the *Task* leaderboard, the senses of the nouns in each question and answer held information valuable enough to allow our models to surpass one of the three proposed baselines. Regarding our bag-of-words model, we will add a metric that penalizes an answer if the senses it displays are wildly different to those of the initial question. This penalty could reduce the impact red herrings typically found in riddles have on the BOW-WSD model's predictive abilities.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of

deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Christiane Fellbaum. 1998. Wordnet: An electronic lexical database. In *International Conference on Lexical Resources and Evaluation*. LREC.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2023. Brainteaser: Lateral thinking puzzles for large language model. *arXiv preprint arXiv:2310.05057*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8758–8765.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6:245–255.

A Appendix

Library	Version
`NumPy`	1.26.1[7]
`NLTK`	3.8.1[8]
`Transformers`	4.35.0[9]
`Scikit-Learn`	1.4.0[10]
`PyTorch`	2.1.0+cu118[11]

Table 3: Table of major Python libraries (and their versions) employed while working to solve the *Task*.

[7] https://numpy.org/doc/

[8] https://www.nltk.org/

[9] https://huggingface.co/docs/transformers/en/index

[10] https://scikit-learn.org/stable/

[11] https://pytorch.org/docs/stable/index.html

hinoki at SemEval-2024 Task 7: Numeral-Aware Headline Generation (English)

Hinoki Crum and **Steven Bethard**
School of Information
University of Arizona
{hinokicrum,bethard}@arizona.edu

Abstract

Numerical reasoning is challenging even for large pre-trained language models. We show that while T5 models are capable of generating relevant headlines with proper numerical values, they can also make mistakes in reading comprehension and miscalculate numerical values. To overcome these issues, we propose a two-step training process: first train models to read text and generate formal representations of calculations, then train models to read calculations and generate numerical values. On the SemEval 2024 Task 7 headline fill-in-the-blank task, our two-stage Flan-T5-based approach achieved 88% accuracy. On the headline generation task, our T5-based approach achieved RougeL of 0.390, BERT F1 Score of 0.453, and MoverScore of 0.587.

1 Introduction

Comprehension of numerical values can significantly enhance performance in certain tasks as numbers provide important information in words. Numerical values are particularly important in accounting and finance fields as the majority of data is in monetary terms. While words can be ambiguous, numbers provide clear and precise information. They not only represent exact numerical values, but can also indicate a magnitude of the subject matter, which can be critical to fully understand a text.

Despite the significance of numerical values, much natural language processing work has treated numerical words in the same manner as all other words, without any direct understanding of the values they represent. As a result, numerical reasoning is still challenging for natural language processing models, even the pre-trained language models that have been so successful on other natural language processing tasks.

NumEval (Chen et al., 2024) provides shared tasks that encourage research systems to generate headlines with accurate numeral information. We fine-tuned pre-trained models for two sub-tasks. In the first, models are required to compute the correct number to fill the blank in a news headline given the corresponding news article. In the second, models are required to construct an entire headline (including its numerical information) based on the provided news article.

2 Related Work

A Math Word Problem (MWP) consists of a short natural language narrative describing a state of the world and poses a question about some unknown quantities Patel et al. (2021). The MWP task is a type of semantic parsing task where given an MWP the goal is to generate an equation, which can then be evaluated to get the answer. The task is challenging because a machine needs to extract relevant information from natural language text as well as perform mathematical reasoning to solve it. Patel et al. (2021) proved in their paper that the existing models can rely on superficial patterns present in the narrative of the MWP and achieve high accuracy without even looking at the question.

Ran et al. (2019) proposed a numerical Machine Reading Comprehension model named NumNet, which utilizes a numerically-aware graph neural network to make numerical comparison and performs numerical reasoning over numbers in the question and passage. Their NumNet model achieved some numerical reasoning ability with Exact Match (EM) of 64.56 and numerically-focused F1 score of 67.97 on the test data. However, NumNet is not applicable when an intermediate number has to be derived in the reasoning process such as from arithmetic operation.

Geva et al. (2020) proposed a general method for injecting additional skills into Language Models, assuming automatic data generation is possible. They applied their approach to the task of numerical reasoning over text, using a general-purpose

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 34–39

model called GENBERT, and a simple framework for generating large amounts of synthetic examples. Their experiments demonstrated the effectiveness of their method, showing that GENBERT successfully learns the numerical skills, and performs on par with similarly sized state-of-the-art numerical reasoning over text models.

Petrak et al. (2023) proposed arithmetic-based pre-training that combines contrastive learning to improve the number representation, and a novel inferable number pre-training objective to improve numeracy. Their experiments showed performance improvements due to better numeracy in three different state-of-the-art pre-trained language models, BART, T5, and Flan-T5, across various tasks and domains, including reading comprehension, inference-on-tables, and table-to-text generation.

Peng et al. (2021) proposed a novel pre-trained model, namely MathBERT, which is the first pre-trained model for mathematical formula understanding. MathBERT was jointly trained with mathematical formulas and their corresponding contexts to evaluate three downstreamtasks, including mathematical information retrieval, formula topic classification and formula headline generation. Formula headline generation is a summarization task aiming to generate a concise math headline from a detailed math question which contains math formulas and descriptions. In addition, in order to further capture the semantic-level structural features of formulas, a new pre-training task is designed to predict the masked formula sub-structures extracted from the Operator Tree (OPT), which is the semantic structural representation of formulas.

3 Data

3.1 Subtask 1: Headline Fill-in-the-Blank

The training dataset (Huang et al., 2023) consists of 21,157 news articles with masked headlines and the validation dataset consists of 2,572 news articles with masked headlines. Both the training and validation datasets have four columns consisting of "news", "masked headline", "calculation" and "answer" as shown in Table 1. The numerical values which should be predicted in the masked headline are shown in underscores. The calculation column shows the operations required to get to the answers, such as copy, round, paraphrase, convert number words to numbers, and arithmetic operations. The calculation may also be a combination of multiple operations.

The test set consists of 4,921 news articles with masked headlines without the calculation and answer columns.

3.2 Subtask 2: Headline Generation

The training dataset consists of 21,157 news articles with unmasked headlines and the validation dataset consists of 2,365 news articles with headlines. The datasets for subtask 2 do not have the calculation column. The test dataset consists of 5,227 news articles.

4 Methodology

4.1 Models

We employed several different types of neural network models for these tasks.

DistilRoBERTa RoBERTa (Liu et al., 2019) is transformer network trained on 16GB of text with a masked language modeling objective, making it appropriate for fill-in-the-blank tasks like Subtask 1. RoBERTa follows the standard transformer formulation, using self-attention to process an input sequence and generate contextualized representations as the output sequence. DistilRoBERTa (Sanh et al., 2019) is a distilled version of the RoBERTa-base model.

T5-Headline-Pleban The Text-to-Text-Transfer-Transformer (T5) model is a transformer network trained on 750GB of text with a language modeling objective where multiple consecutive tokens are masked and the output is a sequence. Because T5 models are designed to produce a sequence, they are suitable for headline generation tasks like Subtask 2. T5-Headline-Pleban (Pleban, 2020) is a T5-base model that was further fine-tuned to predict headlines from articles using a collection of 500k articles.

T5-Title-Zearing (Zearing, 2022) is a T5-base model that was further fine-tuned to predict titles from articels using a collection of Medium articles.

Flan-T5-LaMini Flan-T5 is an enhanced version of T5 that has been finetuned on a mixture of tasks (Chung et al., 2022). LaMini-Flan-T5-783M is a fine-tuned version of google/flan-t5-large on the LaMini-instruction dataset that

news	masked headline	calculation	answer
(Apr 18, 2016 1:02 PM CDT) Ingrid Lyne, the Seattle mom allegedly murdered while on a date, left behind three daughters—and a GoFundMe campaign set up to help the girls has raised more than $222,000 so far, Us reports. A friend of the family set up the campaign, and says that all the money raised will go into a trust for the girls, who are ages 12, 10, and 7. Lyne's date was charged with her murder last week.	$____K Raised for Kids of Mom Dismembered on Date	Paraphrase(222,000,K)	222

Table 1: Sample Data for Subtask 1

contains 2.58M samples for instruction fine-tuning (Wu et al., 2023).

4.2 Subtask 1: Headline Fill-in-the-Blank

We trained three types of models for subtask 1.

4.2.1 DistilRoBERTa

To construct the input for DistilRoBERTa, we concatenated the news text, masked headline, and calculation columns. The underscores we replaced with DistilRoBERTa's mask token, and time stamps were removed. We then trained DistilRoBERTa to predict the answer given this input, using a learning rate of 5e-5. At prediction time, we took the top 20 highest probability vocabulary tokens predicted by the model for the mask token, and returned the first numerical value.

4.2.2 T5 One-Step

To construct the input for our one-step T5 and Flan-T5 models, we replaced the underscores in the masked headline with the token <extra_id_0> and concatenated it to the news text. Unlike DistilRoBERTa, we did not include the calculation in the input as we found it deteriorated model performance. We trained the two T5 models with a learning rate of 5e-5, and the Flan-T5 model with a learning rate of 2e-5. At prediction time, we found the index of the extra token in the model output and used that to extract the numerical value.

4.2.3 T5 Two-Step

As Patel et al. (2021) demonstrated, if models rely on shallow heuristics to solve the majority of math problems without word-order information or question text, instead of training the models to have them directly predict numerical values from question texts, it might be more beneficial to train them

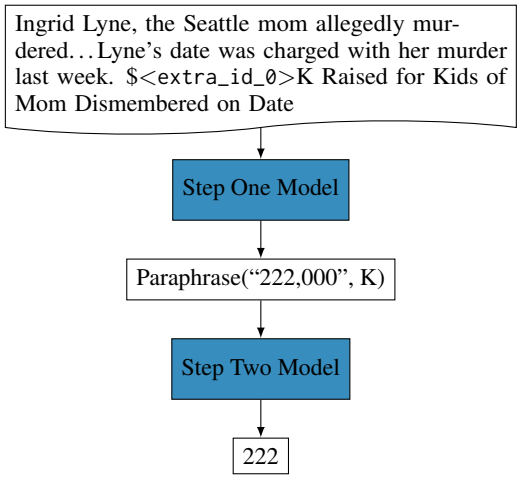


Figure 1: Two-step prediction for the headline fill-in-the-blank task.

to predict numerical values from formulas or calculation methods. Accordingly, we propose two-step models in which we constructed two training sets. For step one, we concatenated the news text and the masked headline as input, and used the calculation as output. For step two, we used the calculation as input, and the answer as output. We then trained two models, one on each dataset. At prediction time, we applied the step-one model to the concatenation of the news text and masked headline, then passed the output of the step-one model as the input to the step-two model, which then predicted the final answer. We used the same extra token processing and learning rates as in the T5 One-Step approach. This process is shown diagrammatically in Figure 1.

4.3 Subtask 2: Headline Generation

We trained T5 models with the news text as input and the headline as output. We prefixed the input

Data	Model	Before	After
Val	DistilRoBERTa	6.23	3.68
Val	T5-Headline-Pleban	2.66	1.05
Val	T5-Title-Zearing	2.14	1.05

Table 2: Perplexity of models on the Headline Fill-in-the-Blank validation data

Data	Model	1 Step	2 Steps
Val	DistilRoBERTa	0.798	N/A
Val	T5-Headline-Pleban	0.877	0.879
Val	T5-Title-Zearing	0.878	0.881
Val	Flan-T5-LaMini	0.886	**0.902**
Test	Flan-T5-LaMini	-	0.88
Test	GPT-3.5 baseline		0.74
Test	Best system		0.95

Table 3: Accuracy of models on the Headline Fill-in-the-Blank validation and test data

with a prompt "headline: " so T5 knows this is a headline generation task. Both T5 models were trained with the learning rate of 5e-5. We also tried Flan-T5, but results were similar to the other T5 models, so we focused our analysis on the headlines generated by the T5 models only.

5 Results and Evaluation

5.1 Subtask 1: Headline Fill-in-the Blank

One measure of the quality of a model is perplexity, defined as the exponential of the cross-entropy loss over the probabilities the model assigns to the next word in all the sentences of the test set. As shown on Table 2, perplexity decreased significantly for all models after training.

A more direct measure of the models in the headline fill-in-the-blank task is accuracy, counting the fraction of times that the model's prediction of a numeric value exactly matched the expected numeric value in the data. Table 3 shows accuracy of the different models on the validation data. Training in two steps did not improve the performance of T5-Headline-Pleban or T5-Title-Zearing, but did slightly improve performance of Flan-T5-LaMini. The final row of Table 3 shows that the best model, two-step Flan-T5-LaMini, achieved 88% accuracy on the test data.

We manually analyzed the errors of the models on the validation data. Errors often revolve around

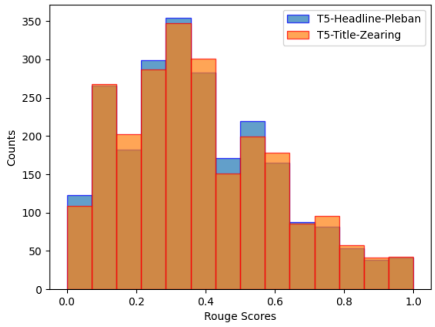


Figure 2: Rouge Scores of models on the Headline Generation validation data

arithmetic operations, rounding of decimal numbers, and the combination of operations. Table 4 shows examples of such errors.

While Patel et al. (2021) achieved about 65% accuracy from their best model, we achieved on the validation dataset the accuracy of 82% on predicting correct formulas while 88% on predicting correct numerical values from those formulas. We also noted that the accuracy on predicting right answers from correctly predicted formulas is 96%. This indicates that the models have no problem with making predictions from simple heuristics, which agrees with the findings by Patel et al.

5.2 Subtask 2: Headline Generation

We evaluated headline generation models based on how well their generated headlines matched the headlines in the data. We used two metrics, Recall-Oriented Understudy for Gisting Evaluation (ROUGE) and BERTScore. Both of these metrics measure the similarity between the predicted headlines and actual headlines, with the former relying on word n-grams and the latter relying on cosine similarity over contextualized embeddings derived from BERT (Mansuy, 2023). Figures 2 and 3 show the distribution of scores of the different T5 models over the validation data. The models are similar in terms of ROUGE score, but T5-Headline-Pleban performs slightly better than T5-Title-Zearing in terms of BERTScore.

We also used the official scoring script, producing the results shown in the first two rows of Table 5, where we see that T5-Title-Zearing is slightly better than T5-Headline-Pleban on the validation data for most measures. We thus submitted T5-Title-Zearing on the test set. The last row of Table 5 shows that it acheived 62.3% numerical accuracy

Actual	Predicted
Round(Divide(268,30),0)	Copy(9)
Round(1.29,0)	Span(a trillion)
Subtract(Sep 5,July 8)	Subtract(30,7)
Add(22,Trans(four))	Add(Trans(four),22)
Subtract(2014,1974)	Subtract(2018,1974)
Multiply(Trans(one-quarter),100)	Multiply(Divide(Trans(one-quarter),100)

Table 4: Examples of incorrect calculations generated by Flan-T5-LaMini on the Headline Fill-in-the-Blank data

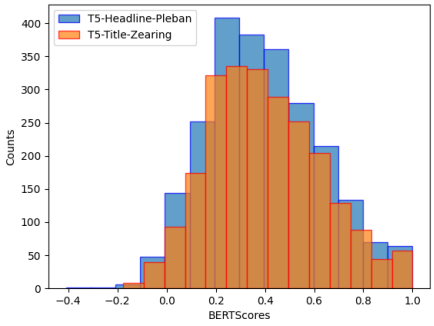


Figure 3: BERTScores of models on the Headline Generation validation data

on the test set with F1 scores of R1 of 43.1, R2 of 19.7 and RL of 40.0. MathBERT which trained with source texts, formulas and OPTs, achieved F1 scores of R1 of 61.25, R2 of 48.06 and RL of 57.72 on formula headline generation, which indicates that training the models with OPTs as inputs help improving the results.

We manually analyzed some of the errors of the models on the validation data. Table 6 shows examples of the headlines generated by T5 models. Items 1 and 2 show that both models properly included the numerical values and captured the meanings, but the expressions of the numerical values and the wordings are different. Several headlines were perfectly generated by T5-Headline-Pleban but not by T5-Title-Zearing, as in item 3, and vice versa, as in item 4. Item 5 is an example of perfect generations by both models. In item 6, a woman who offered a $25K reward for information on her husband's killer was arrested as the killer after 13 years. T5-Headline-Pleban properly captured the $25K reward, but failed to mention that she was the one who got arrested, while T5-Title-Zearing did the opposite. The predictions for item 7 made by both models are close to the actual headline, but the actual headline is designed to better draw attention and drive curiosity. For items 8 and 9, both T5 models failed to capture the appropriate numerical values. Item 10 is an example that both models failed to include any numerical value in the headlines.

6 Conclusion

T5 language models seem capable of generating meaningful headlines including appropriate numerical values. Although the models can reasonably compute the correct numbers from the provided news to fill the blank in headlines, they sometimes failed reading comprehension and arithmetic operations. In hope of overcoming those limitations, we trained them to generate the calculation methods first and then trained again with those calculations as inputs to predict the numerical values to fill the blank in the news headlines, but it did not significantly improve the results. In the future, we plan to try larger pre-trained models, which might improve performance. Also, the training datasets that we used are relatively small. If we increase the data size by data augmentation, we may be able to obtain better results.

References

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex an Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

		Numerical Accuracy			ROUGE			BERTScore			
Data	Model	Overall	Copy	Reasoning	1	2	L	P	R	F1	MoverScore
Val	T5-Headline-Pleban	48.8			43.7	20.1	39.5	46.0	44.8	45.4	59.0
Val	T5-Title-Zearing	51.2			44.1	20.3	39.9	47.3	44.7	46.0	59.2
Test	T5-Title-Zearing	62.3	66.3	55.1	43.1	19.7	40.0	47.2	43.4	45.3	58.7
Test	Best system	73.5	82.2	67.3	48.9	24.7	44.2	51.6	50.1	50.4	60.6

Table 5: Performance of models on the Headline Generation data

	Actual	T5-Headline-Pleban	T5-Title-Zearing
1.	25% of Freed Gitmo Detainees Returned to Terror	1 in 4 Ex-Gatoramo Detainees Linked to Terrorists	1 in 4 Guantánamo Detainees Linked to Terror
2.	3rd Victim Dead in Quarry Shooting; Manhunt Still On	3 Killed in California Quarry Shooting Spree	3 Dead in California Quarry Shooting Spree
3.	Cop Finds Driver Playing Pokemon Go on 8 Phones	Cop Finds Driver Playing Pokemon Go on 8 Phones	Driver Playing Pokemon Go on 8 Phones: Trooper
4.	Tucson Cops Search for Missing Girl, 6	6-Year-Old Missing in Tucson	Tucson Cops Search for Missing Girl, 6
5.	NBC Paid Chelsea Clinton $600K a Year	NBC Paid Chelsea Clinton $600K a Year	NBC Paid Chelsea Clinton $600K a Year
6.	She Offered $25K to Catch a Killer. Now, She's Charged	Woman Offers $25K Reward in Husband's 2006 Murder	13 Years Later, Cops Say She Killed Her Husband
7.	Ex-Congressman Caught With $90K in Freezer Is Guilty	Ex-La. Rep Convicted of Taking $90K in Bribes	Former Louisiana Rep Convicted of Stealing $90K in Bribes
8.	We Drink and Drive an Estimated 121M Times a Year	1 in 5 Adults Admit Driving While Under the Influence	1.8% of US Adults Admit Driving While Impaired
9.	Their Film Ran in 14 Theaters. Then Robert Pattinson Called	Robert Pattinson Leads the 30-something Brothers in Good Time	Robert Pattinson's 'Good Time' Is Just the First 5 Minutes
10.	Alec Baldwin Collects $1.4K Every Time He Plays Trump	Alec Baldwin's Trump Impersonation Is 'Puffs'	Alec Baldwin's Trump Impersonation Is a SNL Sting

Table 6: Examples of headlines predicted by models on the Headline Generation data

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Raphael Mansuy. 2023. Evaluating nlp models: A comprehensive guide to rouge, bleu, meteor, and bertscore metrics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP Models really able to Solve Simple Math Word Problems? *arXiv preprint arXiv:2103.07191*.

Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. MathBERT: A Pre-Trained Model for Mathematical Formula Understanding. *arXiv preprint arXiv:2105.00377*.

Dominic Petrak, Nafise Sadat Moosavi, and Iryna Gurevych. 2023. Arithmetic-based pretraining improving numeracy of pretrained language models. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 477–493, Toronto, Canada. Association for Computational Linguistics.

Michal Pleban. 2020. t5-base-en-generate-headline.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2474–2484, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. Lamini-flan-t5-783m.

Caleb Zearing. 2022. article-title-generator.

T5-Medical at SemEval-2024 Task 2: Using T5 Medical Embeddings for Natural Language Inference on Clinical Trial Data

Marco Siino
Department of Electrical, Electronic
and Computer Engineering
University of Catania
Italy
marco.siino@unipa.it

Abstract

In this work, we address the challenge of identifying the inference relation between a plain language statement and Clinical Trial Reports (CTRs) by using a T5-large model embedding. The task, hosted at SemEval-2024, involves the use of the NLI4CT dataset (Jullien et al., 2023a). Each instance in the dataset has one or two CTRs, along with an annotation from domain experts, a section marker, a statement, and an entailment/contradiction label. The goal is to determine if a statement entails or contradicts the given information within a trial description. Our submission consists of a T5-large model pre-trained on the medical domain. Then, the pre-trained model embedding output provides the embedding representation of the text. Eventually, after a fine-tuning phase, the provided embeddings are used to determine the CTRs' and the statements' cosine similarity to perform the classification. On the official test set, our submitted approach is able to reach an F1 score of 0.63, and a faithfulness and consistency score of 0.30 and 0.50 respectively.

1 Introduction

In experimental medicine, clinical trials are essential because they verify the effectiveness and safety of novel treatments (Giaccone, 2002). Clinical Trial Reports (CTRs) are documents that describe the design and outcomes of a clinical trial and are used to direct patient interventions that are specific to them. But with over 400,000 published CTRs and more coming out each year (Bastian et al., 2010), it is not feasible to manually conduct thorough reviews of all the pertinent literature while developing new treatment procedures. For these reasons, the requirement for technologies that can automatically extract and classify information is always expanding.

With the development of machine and deep learning architectures in recent years, there has been a surge in interest in natural language processing, or NLP. Many efforts have gone into creating algorithms that can automatically identify and categorize text information that is accessible on the internet. In the literature, to perform text classification tasks, several strategies have already been proposed. In the last fifteen years, some of the most successful ones have been based on SVM (Colas and Brazdil, 2006; Croce et al., 2022), on Convolutional Neural Network (CNN) (Kim, 2014; Siino et al., 2021), on Graph Neural Network (GNN) (Lomonaco et al., 2022), on ensemble models (Miri et al., 2022; Siino et al., 2022) and, recently, on Transformers (Vaswani et al., 2017; Siino et al., 2022b).

For example, to address the CTR proposed task, and to enable a higher degree of accuracy and efficiency in individualized evidence-based treatment, Natural Language Inference (NLI) (MacCartney, 2009) provides a viable solution for the large-scale interpretation and retrieval of medical evidence (Sutton et al., 2020). SemEval-2024 Task 2 – Multi-Evidence Natural Language Inference for Clinical Trial Data (NLI4CT) (Jullien et al., 2024) – relies on the NLI4CT dataset[1]. The task is to determine the inference relation between a natural language statement, and a CTR. Inference chains in this drop-off range have to be constructed for a significant fraction of the NLI4CT dataset instances. Furthermore, inference on NLI4CT requires quantitative and numerical reasoning. Research has demonstrated that transformer-based models rely on flimsy heuristics for predictions instead of consistently applying this kind of reasoning (Helwe et al., 2021).

To develop our model, we thought of a two-stage architecture. In the first stage, we used a Sentence Transformer specifically trained on the medical domain. On the generated embeddings, we evaluated a cosine similarity to predict the entailment or con-

[1] https://github.com/ai-systems/nli4ct

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 40–46

tradiction relationship between the two sentences analyzed.

The remainder of the paper is structured as follows. We give some background information on Task 2 hosted at SemEval-2024 in Section 2. Section 3 offers an explanation of the submitted approach. We describe the experimental setup to reproduce our work in Section 4. The outcomes of the formal assignment and certain debates are given in Section 5. We provide our conclusion and suggestions for further research in section 6.

We make all the code publicly available and reusable on GitHub[2].

Figure 1: A sample from the official webpage. Given two trials and a section description, a model has to predict if there is entailment or contradiction with regard to the statement provided.

2 Background

We give some background information on Task 2 hosted at SemEval-2024 in this section. The task is predicated on a set of CTRs, statements, labels, and explanations related to breast cancer that have been annotated by domain experts.

The gathered CTRs are compiled into four components for the textual entailment task:

- *Eligibility criteria* — A list of requirements that patients must meet in order to participate in the clinical trial;
- *Intervention* — Details about the type, strength, frequency, and length of the treatments under investigation;
- *Results* — Units, outcome measures, number of trial participants, and results;
- *Adverse events* — These are the symptoms and indicators that the patients had throughout the clinical study.

With an average length of 19.5 tokens, the annotated statements are sentences that make a claim regarding the data presented in one of the CTR premise's sections. The remarks could compare two CTRs or make assertions about a single CTR. Finding the inference relation (entailment vs. contradiction) between CTR is the problem at hand. The training set provided is identical to the training set used in previous tasks (Jullien et al., 2023b), however, the organizers have performed a variety of interventions on the test set and development set statements, either preserving or inverting the entailment relations. The technical details adopted to perform the interventions were not disclosed, to guarantee fair competition and in the interest of encouraging approaches that are robust and not simply designed to tackle these interventions.

An example is shown in the Figure 1 and is provided in the official task webpage available online[3].

Even if it has already been proved that the Transformers are not necessarily the best option for any text classification task (Siino et al., 2022a), depending on the goal some strategies like domain-specific fine-tuning (Sun et al., 2019; Van Thin et al., 2023), or data augmentation (Lomonaco et al., 2023; Mangione et al., 2022; Siino et al., 2024a) can be beneficial for the considered task.

The training and practice test sets were made available by the task organizers prior to the competition's official commencement. The gold labels were supplied for both sets. Participants could build and test their models during the first phase, called the *practice phase*, by uploading their predictions to CodaLab[4]. The second step, known as the *evaluation phase*, began with the release of the unlabeled test set.

3 System Overview

The rising use of Transformer-based architectures in the literature, has been supported also by several approaches presented at SemEval 2024. These approaches address very different tasks, obtaining interesting results. For example, in the case of the Task 1, where the semantic textual relatedness is evaluated using MPNet (Siino, 2024a), or in the case of the Task 4, where a Mistral 7B model is used for detecting persuasion techniques in meme

[2] https://github.com/marco-siino/SemEval2024/tree/main/Task%202

[3] https://sites.google.com/view/nli4ct/semeval-2024/dataset-description

[4] https://codalab.lisn.upsaclay.fr/competitions/16190

(Siino, 2024c), or, eventually, as in the case of the Task 8, where a DistilBERT model is employed to detect machine-generated text (Siino, 2024b). To develop our model, we also take advantage from a Transformer architecture, creating a two-stage pipeline. In the first stage, we used a *Sentence Transformer* specifically trained on the medical domain. This is a Python framework to create cutting-edge sentence, text, and image embeddings. The initial work is described in (Reimers and Gurevych, 2019). More than 100 languages have sentences and text embeddings that can be computed using this method. Sentences with a similar meaning can subsequently be found by comparing these embeddings, for example, using cosine-similarity. Semantic search, paraphrase mining, and semantic textual similarity can all benefit from this. The framework offers a huge selection of pre-trained models suited for different tasks and is built on PyTorch and Transformers. Moreover, fine-tuning models is also feasible.

The model used as Sentence transformer is T5-large-medical, and it is available on *Hugging Face*[5]. The base model is T5 (Raffel et al., 2020). Specifically, sentences and paragraphs are mapped to a dense vector space of 768 dimensions. PyTorch was used to convert the TensorFlow model st5-large-1 to this one. While the TFHub model and this PyTorch model can provide somewhat different embeddings, they yield the same results when applied to the same benchmarks.

The model was used to map all the words present in the text to the domain-specific embedding. Following the embeddings of the primary section and the statement, the cosine similarity between the two was calculated. In the case of presence of a secondary section, the operation was also carried out between the secondary section and the statement. The cosine similarity between the two embedding vectors is calculated as shown in the Equation 1.

$$\cos(\theta) = \frac{A \cdot B}{\|A\|_2 \|B\|_2} \tag{1}$$

In the first case, if the cosine similarity was greater than 0.5, the label of entailment was assigned, vice versa that of contradiction. In the second case, before calculating the cosine similarity, the average between the cosine similarity score between the two sections and the statement was calculated. Our code is available online together with the predictions generated and sent in relation to the test set.

As noted in the recent study by (Siino et al., 2024b), the contribution of preprocessing for text classification tasks is generally not impactful when using Transformers. More specifically, the best combination of preprocessing strategies does not provide relevant improvements compared to not performing any preprocessing when using Transformers. For these reasons, and to keep our system faster and computationally light, we have not performed any preprocessing on the text.

4 Experimental Setup

We implemented our model on Google Colab[6]. The library we used is Sentence Transformer. The library requires Python[7] (>= 3.8) and PyTorch[8] (>=1.11.0). The dataset provided for all the phases are available on the Official Competition page. On the basis of our preliminary experiments, we found beneficial to set the threshold value for the cosine similarity equal to 0.5. We did perform additional fine-tuning on the T5 embedding. To run the experiment, a T4 GPU from Google has been used. After the generation of the predictions, we exported the results on the JSON format required by the organizers. As already mentioned, all of our code is available on GitHub.

5 Results

For the task the official metric used were F1 (also known as balanced F-score or F-measure), Faithfulness and Consistency.

The F1 score can be described as the harmonic mean of the precision and recall, with a maximum score of 1 and a minimum score of 0. Recall and precision both contribute equally to the F1 score in terms of relative importance. Equation 2 shows the formula for the F1 score.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2}$$

Faithfulness is a measure of the extent to which a given system arrives at the correct prediction for the correct reason. Intuitively, this is estimated by measuring the ability of a model to correctly change its predictions when exposed to a semantic-altering intervention. Given N statements x_i in the

[5] https://huggingface.co/sentence-transformers/sentence-t5-large

[6] https://colab.research.google.com/

[7] https://www.python.org/

[8] https://pytorch.org/

	F1	Faith	Const
T5-large-medical	0.63	0.30	0.50

Table 1: The suggested method's performance on the test set. In the table, the words *Faith* and *Const* stand out for *Faithfulness* and *Consistency*

contrast set (C), their respective original statements y_i, and model predictions *f()* faithfulness can be computed using Equation 3.

$$Faithfulness = \frac{1}{N}\sum_{n=1}^{N} |f(y_i) - f(x_i)| \quad (3)$$

Consistency is a measure of the extent to which a given system produces the same outputs for semantically equivalent problems. Therefore, consistency is measured as the ability of a system to predict the same label for original statements and contrast statements for semantic preserving interventions. That is, even if the final prediction is incorrect, the representation of the semantic phenomena is consistent across the statements. Given N statements x_i in the contrast set (C), their respective original statements y_i, and model predictions *f()* we compute consistency using Equation 4.

$$Consistency = \frac{1}{N}\sum_{n=1}^{N} 1 - |f(y_i) - f(x_i)| \quad (4)$$

In Table 1, the results obtained using the three metrics on the official test set are shown. Considered the very low effort required to run the proposed approach and to generate the predictions, the F1 score of 0.63 appears to be an interesting baseline, while consistency and faithfulness exhibit a very large room for improvements using the proposed approach. It is worth noticing that the approach is a Zero-Shot one with no prior knowledge on the specific task.

In the Table 2, the results obtained by the first three teams and by the last one, as showed on the official CodaLab page, are reported. Compared to the best performing models, our simple approach exhibits some room for improvements. However, it is worth notice that our proposed approach do not require any further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab. We performed few interventions to assess the setup

	F1	Faith	Const
dodoodo (1)	0.78	0.92	0.81
aryopg (2)	0.78	0.95	0.78
jvl (3)	0.78	0.80	0.77
MJ2301 (32)	0.47	0.44	0.47

Table 2: Comparing performance on the test set. In the table are shown the results obtained by the first three users and by the last one. In parentheses is reported the position in the official ranking.

of our approach. For example, we evaluated the number of the epochs to use for fine-tuning the Transformer embedding, the number of warm up steps and the train loss to use. All the details that led our model to reach its final performance, can be deducted from our code available on GitHub.

6 Conclusion

This paper presents the application of T5-large model embedding for addressing the Task 2 at SemEval-2024. For our submission we decided to follow an easy Zero-Shot learning approach, employing as-is, an in-domain pre-trained Transformer. After getting the contextual embedding provided by the Sentence Transformer, we made use of a cosine similarity to calculate the similarity between sentences and generate the entailment/contradiction labels. The task is challenging, and there is still opportunity for improvement, as can be noted looking at the final ranking. Possible alternative approaches include utilizing the zero-shot capabilities of models like GPT, increasing the size of the training set by using further data, or directly integrating ontology-based domain knowledge differently than what has been proposed in our work. To assess the effect of biomedical pre-training on MLMs, performance consistency between sections, generalization capacity of models trained on NLI4CT, performance comparability between numerical and biomedical cases and further error analysis is required. Furthermore, given the interesting results recently provided on a plethora of tasks, also few-shot learning (Wang et al., 2023; Maia et al., 2024; Siino et al., 2023; Meng et al., 2024) or data augmentation strategies (Muftie and Haris, 2023; Tapia-Téllez and Escalante, 2020; Siino and Tinnirello, 2023) could be employed to improve the performance. Eventually, an optimal threshold learnt from the validation dataset could be also employed in future works, in place of the

fixed one that we used in this study. Compared to the best performing models, our simple approach exhibits some room for improvements. However, it is worth to notice that the proposed approach required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab.

Acknowledgments

We would like to thank anonymous reviewers for their comments and suggestions that have helped to improve the presentation of the paper.

References

Hilda Bastian, Paul Glasziou, and Iain Chalmers. 2010. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9):e1000326.

Fabrice Colas and Pavel Brazdil. 2006. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Daniele Croce, Domenico Garlisi, and Marco Siino. 2022. An SVM ensemble approach to detect irony and stereotype spreaders on twitter. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2426–2432. CEUR-WS.org.

Giuseppe Giaccone. 2002. Clinical impact of novel treatment strategies. *Oncogene*, 21(45):6970–6981.

Chadi Helwe, Chloé Clavel, and Fabian M. Suchanek. 2021. Reasoning with transformer-based models: Deep learning, but shallow reasoning. In *3rd Conference on Automated Knowledge Base Construction, AKBC 2021, Virtual, October 4-8, 2021*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and André Freitas. 2023a. NLI4CT: multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 16745–16764. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Francesco Lomonaco, Gregor Donabauer, and Marco Siino. 2022. COURAGE at checkthat!-2022: Harmful tweet detection using graph neural networks and ELECTRA. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 573–583. CEUR-WS.org.

Francesco Lomonaco, Marco Siino, and Maurizio Tesconi. 2023. Text enrichment with japanese language to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2708–2716. CEUR-WS.org.

Bill MacCartney. 2009. *Natural language inference*. Ph.D. thesis, Stanford University, USA.

Beatriz Matias Santana Maia, Maria Clara Falcão Ribeiro de Assis, Leandro Muniz de Lima, Matheus Becali Rocha, Humberto Giuri Calente, Maria Luiza Armini Correa, Danielle Resende Camisasca, and Renato Antonio Krohling. 2024. Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer. *Expert Systems with Applications*, 241:122418.

Stefano Mangione, Marco Siino, and Giovanni Garbo. 2022. Improving irony and stereotype spreaders detection using data augmentation and convolutional neural network. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2585–2593. CEUR-WS.org.

Zong Meng, Zhaohui Zhang, Yang Guan, Jimeng Li, Lixiao Cao, Meng Zhu, Jingjing Fan, and Fengjie Fan. 2024. A hierarchical transformer-based adaptive metric and joint-learning network for few-shot rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(3).

Mohsen Miri, Mohammad Bagher Dowlatshahi, Amin Hashemi, Marjan Kuchaki Rafsanjani, Brij B Gupta,

and W Alhalabi. 2022. Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*, 37(12):11319–11341.

Fuad Muftie and Muhammad Haris. 2023. Indobert based data augmentation for indonesian text classification. In *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, page 128 – 132.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marco Siino. 2024a. All-mpnet at semeval-2024 task 1: Application of mpnet for evaluating semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024b. Badrock at semeval-2024 task 8: Distilbert to detect multigenerator, multidomain and multilingual black-box machine-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024c. Mcrock at semeval-2024 task 4: Mistral 7b for multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2021. Detection of hate speech spreaders using convolutional neural networks. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 2126–2136. CEUR-WS.org.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2022a. Fake news spreaders detection: Sometimes attention is not all you need. *Information*, 13(9):426.

Marco Siino, Marco La Cascia, and Ilenia Tinnirello. 2022b. Mcrock at semeval-2022 task 4: Patronizing and condescending language detection using multi-channel cnn, hybrid lstm, distilbert and xlnet. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 409–417. Association for Computational Linguistics.

Marco Siino, Francesco Lomonaco, and Paolo Rosso. 2024a. Backtranslate what you are saying and i will tell who you are. *Expert Systems*, n/a(n/a):e13568.

Marco Siino, Maurizio Tesconi, and Ilenia Tinnirello. 2023. Profiling cryptocurrency influencers with few-shot learning using data augmentation and ELECTRA. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2772–2781. CEUR-WS.org.

Marco Siino and Ilenia Tinnirello. 2023. Xlnet with data augmentation to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2763–2771. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2022. T100: A modern classic ensemble to profile irony and stereotype spreaders. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2666–2674. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024b. Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

Reed T. Sutton, David Pincock, Daniel C. Baumgart, Daniel C. Sadowski, Richard N. Fedorak, and Karen I. Kroeker. 2020. An overview of clinical decision support systems: benefits, risks, and strategies for success. *npj Digit. Medicine*, 3.

José Medardo Tapia-Téllez and Hugo Jair Escalante. 2020. Data augmentation with transformers for text classification. In *Advances in Computational Intelligence*, pages 247–259, Cham. Springer International Publishing.

Dang Van Thin, Duong Ngoc Hao, and Ngan Luu-Thuy Nguyen. 2023. Vietnamese sentiment analysis: An overview and comparative study of fine-tuning pretrained language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12):7789–7802.

CTYUN-AI at SemEval-2024 Task 7: Boosting Numerical Understanding with Limited Data Through Effective Data Alignment

Yuming Fan* and **Dongming Yang** * † and **Xu He**
fanym@chinatelecom.cn , yangdongming@pku.edu.cn,
hex30@chinatelecom.cn,
China Telecom
Cloud Technology Co., Ltd

Abstract

Large language models (LLMs) have demonstrated remarkable capabilities in pushing the boundaries of natural language understanding. Nevertheless, the majority of existing open-source LLMs still fall short of meeting satisfactory standards when it comes to addressing numerical problems, especially as the enhancement of their numerical capabilities heavily relies on extensive data. To bridge the gap, we aim to improve the numerical understanding of LLMs by means of efficient data alignment, utilizing only a limited amount of necessary data. Specifically, we first use a data discovery strategy to obtain the most effective portion of numerical data from large datasets. Then, self-augment is performed to maximize the potential of the training samples. Thirdly, answers of all training samples are aligned based on some simple rules. Finally, our method achieves the first place in the competition, offering new insights and methodologies for numerical understanding research in LLMs.

1 Introduction

In recent years, the field of Natural Language Processing (NLP) has witnessed remarkable advancements, particularly with the advent of generative large language models (Jiang et al., 2023; Bai et al., 2023; Yang et al., 2023; Brown et al., 2020). These models have predominantly focused on textual data, demonstrating impressive capabilities in understanding and generating human-like text. However, an often-overlooked aspect of these developments is the nuanced role that numerical data plays in fully grasping the semantics of language. This oversight becomes particularly glaring in specialized fields such as stock market analysis, medical diagnostics, and legal decisions (Cortis et al., 2017; Modi et al., 2023; Jullien et al., 2023). In these domains, subtle numerical differences can have far-reaching implications, significantly affecting outcomes and decisions. Thus, the ability to understand and work with numbers, in these contexts underscores a critical gap in the semantic understanding capabilities of current language models.

Acknowledging this deficiency, there has been a growing interest within the NLP community towards enhancing the textual numeracy and computational abilities (Huang et al., 2023) of language models. This burgeoning interest has culminated in the introduction of SemEval2024's Shared Task 7. This innovative task is strategically designed to elevate the standards in the field by promoting the development of models that excel not only in literacy but also in computational skills. Such models promise to significantly boost usefulness and efficiency across a wide array of applications, ranging from automated financial analysis to predictive healthcare diagnostics and beyond.

However, the enhancement of numerical capabilities of LLMs heavily relies on the inclusion of a large amount of data, posing two significant challenges. On one hand, obtaining high-quality numerical annotated data is costly, as it requires significant economic costs and manual effort from professional annotators. On the other hand, the extensive use of as much data as possible to train the model can diminish the utility of high-value data and lead to increased computational consumption.

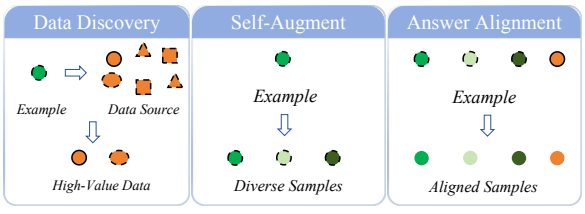


Figure 1: Pipeline of our system in this task.

As illustrated in Figure 1, this paper focuses on how to efficiently use a limited amount of data to

*Equal Contribution.
†Corresponding Author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 47–52
June 20-21, 2024

improve the numerical capabilities of LLMs. Our goal is to enhance the model's performance in numeral while minimizing costs, such as data annotation expenses and computational requirements for model training. Our method propose effective data alignment by employing strategies of data discovery, self-augment and answer alignment. The contributions are summarized as follows:

- A strategy of data discovery is proposed to extract numerical training samples, obtaining the most effective portion of numerical data from datasets and minimizing training costs.
- We implement original self-augment to all the training samples to maximize their effectiveness in enhancing the numerical capabilities of LLM.
- We align answers of all training samples according some customized rules to improve LLM's numerical reasoning performance and shorten the reasoning path.

After conducting numerous experiments and iterating on our strategies, we are proud to announce that we have secured the championship title at the competition of SemEval-2024 Task 7. Detailed ablation study and analysis of our method are also provided in this paper to identify contributions from individual components and facilitate future research.

2 Background

GPT-3 (Brown et al., 2020) marked significant progress in large language models, enhancing few-shot learning and demonstrating robustness across diverse NLP datasets. Bai et al. (Bai et al., 2023) developed the qwen model, notable for its performance in various tasks, particularly its chat model refined through human feedback. However, these models largely focus on textual data, paying limited attention to the importance of numerical values in semantic understanding.

Addressing this, the NumHG (Huang et al., 2023) dataset was introduced, focusing on generating news headlines with numerical information. Evaluations of high-performing models indicated room for improvement in numerical accuracy, aiming to advance research in numerically-focused headline generation and improve task performance.

Additionally, learning Mathematical Reasoning for tasks like GSM8K (Cobbe et al., 2021) and MetaMATH (Yu et al., 2023) remains a significant challenge for LLMs. Enhancing LLM reasoning through augmented output sequences (Wei et al., 2022) has been explored, with methods like Complexity-based CoT (Fu et al., 2022) showing that increased in-context steps can improve performance. Self-Consistency approaches (Wang et al., 2022) use multiple reasoning paths and majority voting to select answers. Other works leverage closed-source LLMs (Brown et al., 2020) for knowledge distillation (Magister et al., 2022), while some apply rejection sampling for better reasoning (Yuan et al., 2023). Techniques like the reinforced evol-instruct method (Luo et al., 2023) and constraint alignment loss for calibration (Wang et al., 2023) also contribute to the advancement of LLMs in mathematical reasoning.

Building on these developments, our work introduces a novel approach to refine LLMs' numerical reasoning capability. We fine-tune our base model with a curated selection of numerically samples, focusing on diversity and efficiency to cover a broader range of mathematical concepts.

3 System Overview

In this section, we will introduce our proposed method from several aspects. We start with data analysis of SemEval-2024 Task 7. Then, we present the proposed data discovery, self-augment and alignment strategies.

3.1 Data Analysis

The competition dataset, NumHG, provided news articles with headlines, where the task involved identifying masked numerical values in the headlines and explaining the calculations behind these numbers. Each data sample from NumHG comprises four elements: News, masked headline, calculation, and answer. As shown in Table 1, we conducted an analysis of the mathematical processing utilized in each data sample, and discovered the following: (1) Most answers can be directly copied from the text, indicating that these numerical values are explicitly mentioned. (2) Additionally, a portion of the answers required converting textual descriptions into numerical forms, involving text understanding and translation. (3) Simple mathematical operations, such as basic arithmetic and rounding, are also involved in a small subset of the dataset, demanding LLM to perform context-based mathematical operations.

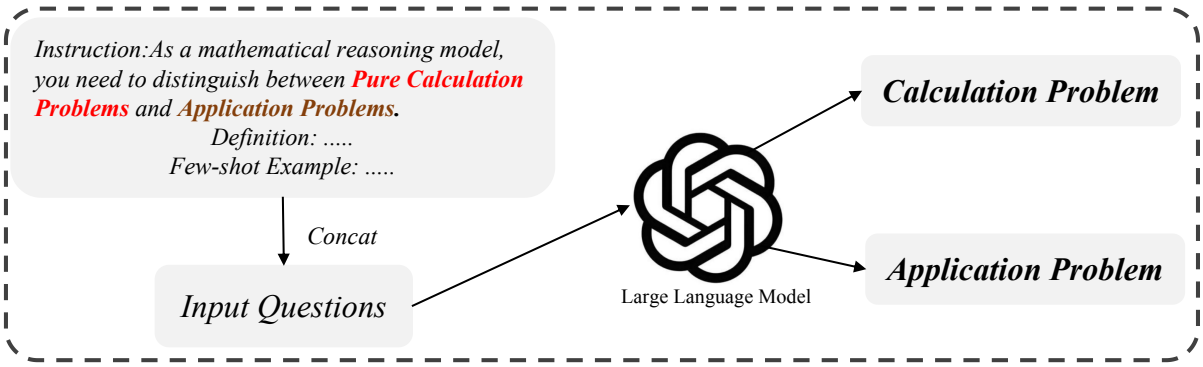


Figure 2: Data Discovery: Demonstrates the selection and integration of applicable problems from GSM8K and MetaMathQA datasets into the training set.

Example 3.1 Application Problem

At Rosa's Rose Shop, a bouquet containing a dozen roses costs $\$20$. If the price of a bouquet is directly proportional to the number of roses it contains, A bouquet of ____ roses will cost 65 dollars. What is the value of unknown variable ____?

Example 3.2 Pure Calculation Problem

Factor completely: $x^6 - 3x^4 + 3x^2 - ____$. The answer is 3. What is the value of unknown variable ____?

What is the value of ____ \times (7 - 5) - 5$? The answer is 1. What is the value of unknown variable ____?

Figure 3: Example of mathematical application problems and pure calculation problems in our dataset.

Table 1: Analysis on mathematical processing in the NumHG dataset.

Mathematical Processing Type	Count
Copy	15998
Trans	4111
Paraphrase	1727
Round	716
Subtract	496
Add	408
Span	104
Multiply	81
Divide	51
SRound	37

The above analysis reveals that the dataset and task possess distinct characteristics (e.g., emphasize understanding numbers within text rather than solving complex mathematical problems), suggesting that limited relevant data could potentially aid in enhancing performance. Furthermore, the high similarity among samples in this dataset underscores the importance of effective data augmentation and alignment strategies to maximize the utility of training samples.

To undertake the aforementioned investigation, we employed Qwen-Chat (Bai et al., 2023) as our base model, setting the input as a concatenation of news and masked headline, and output as a combination of calculation and answer to compile our training set. We crafted a preset prompt, *You are a numerical reasoning model. Please compute the correct number to fill the blank in a news headline.*, to utilize the inherent command-following ability of the LLM.

3.2 Data Discovery

As mentioned previously, we advocate for extracting a limited yet most effective subset of the dataset to enhance model performance. Specifically, we utilized the GSM8K and MetaMathQA datasets as the complementary source of external training data.

It is noted that, deviating from standard math tasks, the NumHG dataset focuses on understanding numerical semantics rather than complex calculations, primarily involving basic arithmetic and sourced from real news. Thus, we first integrated the GSM8K samples and selectively utilized MetaMathQA samples relevant to variable X to form a new collection of numerical samples, matching the competition's focus on masked numbers. Then, incorporating the analysis of both the NumHG dataset and general mathematical datasets (i.e., GSM8K and MetaMathQA), we have defined all mathematical samples into two categories: addressing mathematical application problems and pure

calculation problems. Specifically, we utilized a large language model's few-shot learning to classify the collected samples. We initially crafted several examples for each problem type as input to guide the mode, which covered a broad spectrum of scenarios to enrich the model's adaptability. Secondly, we instructed the model to distinguish between the given samples, categorizing them as either application or pure calculation problems, as illustrated in Figure 2.

Figure 3 shows examples of mathematical application problems and pure calculation problems. Ultimately, we rely on the model's output to determine the category of the input questions. We found that around 78% of GSM8K questions are application-based, versus 23% in MetaMathQA, aiding our understanding of each dataset's distribution and shaping our data Strategies. Finally, instead of using all external samples, we only retained the numerical-based and application problem samples as supplementary training data. This allows us to maximize the improvement in model performance using as little data as possible.

3.3 Data Self-Augment

Given the high similarity among samples in the NumHG dataset, we try to improve the diversity of samples and the difficulty of the task through data augmentation. Inspired by strategies from the visual domain (Jo and Yu, 2021), we introduced sentence-level random shuffling as a data self-augment strategy, as shown in figure 4. Our goal is to generate structurally diverse training samples while preserving the core information of texts. After reshuffling sentences within each training sample, the LLM continues to perform the original task of filling numerical values across structurally varied texts.

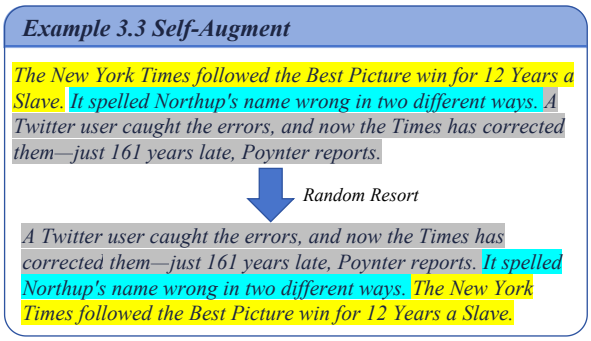


Figure 4: Self-Augment Strategy: sentence-level random shuffling to increase sample diversity while preserving key numerical information.

Although this strategy may disrupt the coherence between sentences within each sample, our experiments have found that it effectively improved the model's ability to handle mathematical problems in more complex contexts.

3.4 Answer Alignment

Adhering to the shared task submission system, the model's output should be a string convertible to a numerical value, devoid of computational methods and descriptive characters. Incorporating the requirements above, we further devised a strategy to simplify the model's output, enhancing model's numerical reasoning performance and shorten the reasoning path. Another reason for implementing this strategy was to ensure data consistency without compromising performance, given the unique characteristics of the competition's computational expressions such as 'Copy and Add'.

Hence, for all samples (i.e., from NumHG, GSM8K and MetaMathQA), we employed regular expressions to directly extract the numerical value as the output for training. Figure 5 shows some example from NumHG. Additionally, this strategy reduced cognitive inference time on 4,921 test instances by eliminating complex computational steps, offering a more direct feedback path.

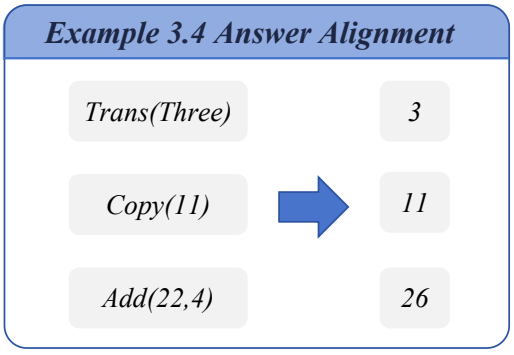


Figure 5: Examples of aligning answers in NumHG, simplifying the model's inference path.

4 Experimental Setup

We utilized the Qwen-72B-Chat model as our base model, on which we performed full parameter fine-tuning. The experiments were conducted on Nvidia A800 GPU with 80GB of memory. During training, we set the maximum token length to 2048, batch size to 8, and performed gradient accumulation every step. An initial learning rate of 5e-6 was set, employing a cosine decay strategy, for a total

of 3 epochs of training. All samples processed by our strategies were used as training data, while the validation set of NumHG were utilized for error analysis. For inference, we employed the default inference parameters.

5 Results

5.1 Final Result

Team	Private Score	Public Score
CTYUN-AI	0.95	0.94

Table 2: Competition results of our team.

At the NumEval-2024 Task 7, the public and private scores were derived from 20% and 80% of the test set data, respectively. In the final standings among all teams, we secured the first place with a private score of 0.95, as shown in Table 2.

This achievement highlights the effectiveness of our method, especially in the more heavily weighted portion of the test set.

5.2 Ablation Study

We took ablation studies to confirm each strategy's (i.e., Data Discovery, Self-Augment, and Answer Alignment) contribution to our final method's success. As shown in Table 3, data discovery, self-augment, and answer alignment brought performance gains of 2%, 4%, 2% separately. Finally, we achieved a 9% performance increase over the baseline in total, underscoring the significance of our approach against high benchmarks.

Method	Private Score	Public Score
Base Data	0.86	0.87
Base Data w/ Prompt	0.87	0.88
+ Data Discovery	0.89	0.88
+ Self-Augment	0.93	0.91
+ Answer Alignment	0.95	0.94

Table 3: Ablation study on our method.

Meanwhile, we evaluated the impact of using samples for mathematical application versus pure calculation problems during data discovery, as shown in Table 4. Findings show application-type problems improve model performance by enhancing real-world numerical understanding, while pure calculation samples negatively affect it due to their complexity leading to intricate computations.

Method Type	Method	Private Score	Public Score
Base Data	w/o Prompt	0.86	0.87
	w/ Prompt	0.87	0.88
w/ Data Discovery	Pure Calculation	0.87	0.87 (-0.01)
	Application	0.89 (+0.02)	0.88

Table 4: Ablation study on data discovery and fusion.

5.3 Error Analysis

In analyzing error cases, we discovered that rounding could lead to misunderstandings in numerical comprehension. For instance, one example states: *"...Nielsen numbers show that 31.1 million people,..."* hence the answer to the question *"____M Watched Jackson Memorial"* should be 31.1, yet the model predicted 31. This indicates that the model's rounding may lead to incorrect answers.

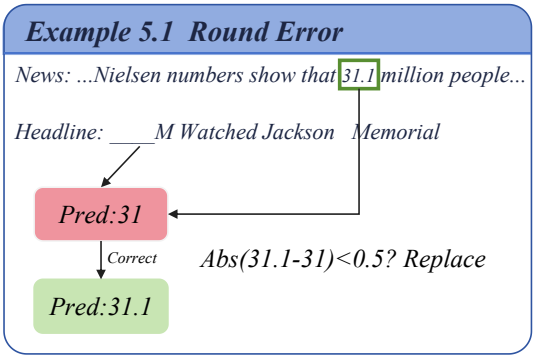


Figure 6: A rounding error case and we introduce a post-processing strategy for it.

As shown in Figure 6, we implemented a post-processing strategy to correct rounding errors. This involves extracting all numbers from the text, comparing them with the model's prediction, and adjusting predictions within a 0.5 difference to the nearest number. This method enhanced our test set performance to 0.95, although it was not included in the competition submission.

6 Conclusion

In this work, we have demonstrated an approach to enhance numerical understanding in large language models (LLMs) using limited data through effective data alignment. Our method integrated data discovery, self-augment, and answer alignment strategies, and significantly improved the model's performance on numerical reasoning tasks. Our success in SemEval-2024 Task 7 highlights the potential of our method in advancing natural language processing, particularly for enhancing the various basic capabilities of large language models.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, and et al. 2023. Qwen technical report. *arXiv:2309.16609*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and et al. 2020. Language models are few-shot learners.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 519–535.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Sanghyun Jo and In-Jae Yu. 2021. Puzzle-cam: Improved localization via matching partial and full features. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 639–643. IEEE.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Donal Landers, and André Freitas. 2023. Semeval-2023 task 7: Multi-evidence natural language inference for clinical trial data. *arXiv preprint arXiv:2305.02993*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*.

Ashutosh Modi, Prathamesh Kalamkar, Saurabh Karn, Aman Tiwari, Abhinav Joshi, Sai Kiran Tanikella, Shouvik Kumar Guha, Sachin Malhan, and Vivek Raghavan. 2023. Semeval 2023 task 6: Legaleval–understanding legal texts. *arXiv preprint arXiv:2304.09548*.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. 2023. Making large language models better reasoners with alignment. *arXiv preprint arXiv:2309.02144*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

McRock at SemEval-2024 Task 4: Mistral 7B for Multilingual Detection of Persuasion Techniques in Memes

Marco Siino
Department of Electrical, Electronic
and Computer Engineering
University of Catania
Italy
marco.siino@unipa.it

Abstract

One of the most widely used content types in internet misinformation campaigns is memes. Since they can readily reach a big number of users on social media sites, they are most successful there. Memes used in a disinformation campaign include a variety of rhetorical and psychological strategies, including smearing, name-calling, and causal oversimplification, to achieve their goal of influencing users. The shared task's objective is to develop models for recognizing these strategies solely in a meme's textual content (Subtask 1) and in a multimodal context where both the textual and visual material must be analysed simultaneously (Subtasks two and three). In this paper, we discuss the application of a Mistral 7B model to address the Subtask one in English about finding the persuasive strategy that a meme employs from a hierarchy of twenty based just on its textual content. Only a portion of the reward is awarded if the technique's ancestor node is chosen. This classification issue is multilabel hierarchical. Our approach based on the use of a Mistral 7B model obtains a Hierarchical F1 of 0.42 a Hierarchical Precision of 0.30 and a Hierarchical Recall of 0.71. Our selected approach is able to outperform the baseline provided for the competition.

1 Introduction

When information is intentionally crafted to serve a predetermined agenda, we often classify it as propaganda (Geissler et al., 2023). Propaganda employs various psychological and rhetorical strategies to achieve its goals (Çakmak, 2023). These methods encompass the utilization of logical fallacies and the manipulation of audience emotions (Soares et al., 2023). Logical fallacies can be particularly deceptive as they may initially appear sound and impartial, yet upon closer examination, it becomes evident that the conclusion cannot be logically derived from the premises. Another tactic involves employing emotionally charged language to sway the audience's opinion, bypassing rational analysis in favour of an emotional connection. Memes, typically comprising images overlaid with text, serve as a platform for propagandistic dissemination. Within deceptive memes, images either reinforce or complement textual techniques, or they themselves convey persuasive strategies.

To address these objectives, there is an ongoing demand for automated tools capable of extracting and categorizing data from online sources, facilitating the response to both established and emerging societal concerns. Recent advancements in machine and deep learning architectures have spurred heightened interest in Natural Language Processing (NLP). Substantial endeavours have been directed towards devising techniques for the automated identification and categorization of textual content accessible on the internet today. In the literature, to perform text classification tasks, several strategies have already been proposed. In the last fifteen years, some of the most successful strategies have been based on SVM (Colas and Brazdil, 2006; Croce et al., 2022), on Convolutional Neural Network (CNN) (Kim, 2014; Siino et al., 2021), on Graph Neural Network (GNN) (Lomonaco et al., 2022), on ensemble models (Miri et al., 2022; Siino et al., 2022) and, recently, on Transformers (Vaswani et al., 2017; Siino et al., 2022b).

The surge in the adoption of Transformer-based architectures within academic research has been further propelled by diverse methodologies showcased at SemEval 2024. These methodologies address a range of tasks and yield notable outcomes. For instance, in Task 2 (Jullien et al., 2024), T5 is utilized to confront the challenge of identifying the inference relation between plain language statements and Clinical Trial Reports (Siino, 2024b). In Task 10, a Mistral 7B model is employed to perform emotion Recognition in Conversation (ERC) within Hindi-English code-mixed conversations

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 53–59

(Siino, 2024c). Additionally, in Task 8 (Wang et al., 2024), a DistilBERT model is leveraged to identify machine-generated text (Siino, 2024a).

Finally, for the Task 4 at SemEval 2024 (Dimitrov et al., 2024) – Multilingual Detection of Persuasion Techniques in Memes – three Subtasks were proposed. As already stated, in a disinformation campaign, memes effectively manipulate users through various rhetorical and psychological strategies, including causal oversimplification, name-calling, and smear tactics. The objective of this shared task is to develop models capable of detecting these techniques within the textual content of memes alone (Subtask 1), as well as within a multimodal framework where both textual and visual elements are analysed jointly (Subtasks 2 and 3). To face with the first Subtask in English, we proposed a Transformer-based approach which made use of Mistral 7B (Jiang et al., 2023). We used the model in a particular few-shot way described in the rest of this paper. Specifically, we provided the definitions of the 20 techniques to the model to identify, given each sample, all the techniques detected. We opted for Mistral 7B because the comparative analysis between Mistral 7B and other leading models, namely Llama 2 and Llama 1, reveals noteworthy advancements in common NLP tasks. Across multiple benchmark evaluations, Mistral 7B consistently exhibits superior performance in comparison to Llama 2, a prominent open 13B model. Moreover, its efficacy extends beyond mere parity with, but rather exceeds, the achievements of Llama 1, a state-of-the-art 34B model, particularly in tasks pertaining to reasoning, mathematics, and code generation. These findings underscore Mistral 7B's substantive contributions to the advancement of NLP, suggesting its potential as a benchmark model in the field.

The rest of the paper is made as follows. In Section 2 we provide some background on the Task 4 hosted at SemEval 2024. In Section 3 we provide a description of the models presented. In Section 4 we provide details about the experimental setup to replicate our work. In Section 5, the results of the official task and some discussions are provided. In section 6 we present our conclusion and proposals for future works.

We make all the code publicly available and reusable on GitHub[1].

[1] https://github.com/marco-siino/SemEval2024/

2 Background

This section furnishes background information regarding Task 4 (Subtask 1), held at SemEval 2024. The task entails identifying, based solely on the textual content of a meme, which of the 20 persuasion techniques, organized hierarchically, are employed. The selection of an ancestor node of a technique warrants only partial reward. The task thus presents a hierarchical, multilabel classification challenge. The hierarchical structure is illustrated in the official task's page, with 22 techniques depicted, although "Transfer" and "Appeal to Strong Emotion" are excluded from Subtask 1. For comprehensive details, please refer to provided resources.

For all Subtasks, the annotations from the PTC corpus, comprising over 20,000 sentences, were utilized where feasible. Although the corpus pertains to news articles, annotations adhere to identical guidelines, albeit with fewer techniques considered. As highlighted by the task coordinators, certain meme content may be deemed offensive or excessively potent by certain audiences. A similar multilingual corpus was also accessible during SemEval 2023 (Piskorski et al., 2023). Here again, the corpus revolves around news articles across nine languages, yet the number of techniques and annotation guidelines differ marginally. A training set for local system development was additionally provided. Furthermore, the organizers furnished a development set and a public leaderboard for real-time result sharing among task participants. Ultimately, the organizers supplied a test set devoid of annotations and an online submission platform to evaluate the system performance.

Subtask 1 relies on the textual content extracted from memes as input data. Training, development, and test sets for all Subtasks are disseminated as JSON files, with each Subtask having its own individual file. For Subtasks 2a and 2b, in addition to the meme's textual content, input data includes the meme's image. In the Figure 1, is reported a sample from the official competition website[2].

Given the Figure 1:

- *ID* consists of a unique identifier of the example across all the Subtasks;
- *text* represents the textual expression within the meme, formatted as a singular UTF-8

[2] https://propaganda.math.unipd.it/semeval2024task4/

Subtask 1

The entry for that example in the json file for subtask 1 is


```
{
        "id": "125",
        "text": "I HATE TRUMP\n\nMOST TERRORIST DO",
        "labels": [
                "Loaded Language",
                "Name calling/Labeling"
],
"link": "https://..."
},
```


Figure 1: An example from the dataset. In this case, two labels are assigned to the sample's text.

string. Initially, this text is automatically extracted from the meme, subsequently undergoing manual post-processing to rectify errors and arrange it such that each sentence occupies a distinct row. Furthermore, segments of text originating from distinct regions within the image are demarcated by blank rows. Notably, Task 1 qualifies as an NLP endeavour, given that image input is absent;

- *labels* denotes a compilation of permissible technique names identified within the text. These labels serve as the gold standard and will solely be furnished for the training set. In this particular instance, two techniques were identified: "Loaded Language" and "Name calling/Labelling."

3 System Overview

Even if it has already been proved that the Transformers are not necessarily the best option for any text classification task (Siino et al., 2022a), depending on the goal some strategies like domain-specific fine-tuning (Sun et al., 2019; Van Thin et al., 2023), or data augmentation (Lomonaco et al., 2023; Mangione et al., 2022) can be beneficial in several applications.

Our approach is a few-shot one (Littenberg-Tobias et al., 2022) and make use of the above-mentioned Mistral 7B. Mistral 7B, a language model equipped with 7 billion parameters, is designed to excel in both performance and efficiency. Compared to the leading open 13B model (Llama 2), Mistral 7B demonstrates superior performance across all evaluated benchmarks. Moreover, it outperforms the top released 34B model (Llama 1) in tasks related to reasoning, mathematics, and code generation. The model leverages grouped-query attention (GQA) to expedite inference, along with sliding window attention (SWA) to efficiently process sequences of varying lengths while minimizing inference costs. Additionally, a fine-tuned variant, Mistral 7B – Instruct, tailored for adhering to instructions, surpasses the Llama 2 13B – chat model across both human and automated benchmarks. The introduction of Mistral 7B Instruct underscores the ease with which the base model can be fine-tuned to achieve notable performance enhancements. Notably, this variant lacks any moderation mechanisms. The Mistral 7B Instruct variant requires a specific input format, as stated below:

> <s>[INST] Instruction [/INST] Model answer</s>[INST] Follow-up instruction [/INST]

Instruction, along with the following *Model answer*, can be a single sample with the related label or a set of sample/label pairs (realizing, in this case, a few-shot use of the model). Then, *Follow-up instruction* is the current sample for which the prediction has to be provided by the model. More specifically, given the 20 persuasion techniques in memes, we have prepared a text string containing the techniques and their definitions to provide context in the template ready. The definitions of the 20 techniques are provided by the task organizers[3]. At this point, the full text containing the twenty definitions plus the sample to be classified were provided as prompt to Mistral.

Then the question provided as prompt to mistral was: *"Given the above Definitions of the Persuasion Techniques, Identify the Persuasion Techniques used in the Sentence. Answer using ONLY one or more numbers in the range 1-20 separated by commas. No text nor other options are allowed."*.

To this request, the model replied with one or more techniques detected in the corresponding sample. So, as an example, to the sentence: "Happy April Fools Day - - Ooop I mean: March Fools day" the model replied to the prompt with the numbers 3 and 5. These two numbers correspond to the technique 3 (i.e., *Whataboutism*) and to the technique 5 (i.e., *Obfuscation, Intentional vagueness, Confusion*). It is important to mention that we also tried to use the model in a zero-shot configuration. In this case, we just asked the model to pick one or more categories given a meme. Unfortunately, the model did not report one or more correct categories, while developing discussions as answers.

Finally, we collected all the predictions provided

[3] https://propaganda.math.unipd.it/semeval2024task4/definitions.html

on the test set to into a JSON file with the required format to submit our predictions.

As noted in the recent study by (Siino et al., 2024b), the contribution of preprocessing for text classification tasks is generally not impactful when using Transformers. More specifically, the best combination of preprocessing strategies is not very different from performing no preprocessing at all in the case of Transformers. For these reasons, and to keep our system fast and computationally light, we have not performed any preprocessing on the text. The low impact of the best preprocessing techniques - or combinations of techniques - using Transformers, as reported in the study, is due to several factors like preserving the quantity and the quality of the original information available.

4 Experimental Setup

We implemented our model on Google Colab. The library we used comes from Hugging Face and as already mentioned is Mistral 7B[4]. We employed the v0.2 iteration of Mistral 7B, which represents an enhanced version of the Mistral-7B-Instruct-v0.1 model. To harness the capabilities of instruction fine-tuning, prompts must be enclosed within [INST] and [/INST] tokens. Additionally, the initial instruction should commence with a sentence identifier. The next instructions should not. The assistant generation will be ended by the end-of-sentence token ID. We also imported the Llama library (Touvron et al., 2023) from *llama_cpp*. The library is fully described on GitHub[5]. The dataset provided for all the phases are available on the Official Competition page. We did not perform any additional fine-tuning on the model. To run the experiment, a T4 GPU from Google has been used. After the generation of predictions, we exported the results on the format required by the organizers. As already mentioned, all of our code is available on GitHub.

5 Results

The evaluation was done by submitting to the leaderboard the predictions provided by the model. Subtask 1 and 2a are reliant on a hierarchical structure. The gold label consistently corresponds to a leaf node within the Directed Acyclic Graph (DAG). However, any node within the DAG can serve as a predicted label:

- If the prediction does not correspond to a leaf node and is an ancestor of the correct gold label, a partial reward is issued, with the reward magnitude contingent upon the distance between the two nodes. For instance, if the gold label is "Red Herring" and the predicted label is "Distraction" or "Appeal to Logic."

- If the prediction does not align with any ancestor node of the correct label, no reward is granted. For instance, if the gold label is "Red Herring" and the predicted label is "Black and White Fallacy" or "Appeal to Emotions." A graphical representation illustrating this concept is provided.

However, it's worth noting that the hierarchical structure can be disregarded by confining predictions solely to technique names. This approach renders the task analogous to SemEval 2023 Task 3 (Piskorski et al., 2023).

An illustrative example of the evaluation function can be accessed online[6]. In this case, the Subtask consists of a hierarchical multilabel classification task. Drawing from the aforementioned figure depicting the hierarchy, any node within the DAG can be designated as a predicted label. The gold label consistently corresponds to a leaf node within the DAG. Hierarchical-F1, detailed in (Kiritchenko et al., 2006), is employed as the official evaluation metric.

In the Table 1, the results obtained by the first three teams and by the last one, as showed on the official page[7], are reported. Compared to the best performing models, our simple approach exhibits some room for improvements, although it is able to outperform the baseline. However, it is worth notice that it required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab.

6 Conclusion

This paper presents the application of Mistral 7B-model for addressing the Task 4 at SemEval 2024.

[4] https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF

[5] https://github.com/ggerganov/llama.cpp

[6] https://propaganda.math.unipd.it/semeval2024task4/data/hierarchy_evaluation.html

[7] https://propaganda.math.unipd.it/semeval2024task4/SemEval2024task4_test.html

	H-F1	H-Prec	H-Recall
914isthebest (1)	0.752	0.684	0.836
BCAmirs (2)	0.698	0.668	0.732
OtterlyObsessedWithSemantics (3)	0.697	0.648	0.755
Mistral 7B (30)	0.42	0.30	0.71
BASELINE (31)	0.369	0.477	0.300
IIMAS1UTM1LaSalle (33)	0.199	0.755	0.115

Table 1: Comparing performance on the test set for Subtask 1 in English. In the table are shown the results obtained by the first three users and by the last one. Furthermore, is included the result of the baseline considered and of our approach making use of Mistral 7B. In parentheses is reported the position in the official final ranking.

For our submission, we decided to follow a few-shot learning approach, employing as-is, an in-domain pre-trained Transformer. After several experiments, we found beneficial to build a prompt containing the definitions of the techniques in memes. Then we provide, as a prompt, the definitions together with a sample. The model was asked to select all the techniques detected in the sentence. The task is challenging, and there is still opportunity for improvement, as can be noted looking at the final ranking. Possible alternative approaches include utilizing the zero-shot capabilities of other models like GPT and T5, increasing the size of the training set by using further data, or directly integrating ontology-based domain knowledge differently than what has been proposed in our work. Further improvements could be obtained with a fine-tuning and modelling the problem as a different text classification task. Furthermore, given the interesting results recently provided on a plethora of tasks, also other few-shot learning (Wang et al., 2023; Maia et al., 2024; Siino et al., 2023; Meng et al., 2024) or data augmentation strategies (Muftie and Haris, 2023; Siino et al., 2024a; Tapia-Téllez and Escalante, 2020; Siino and Tinnirello, 2023) could be employed to improve the results. Looking at the final ranking, our simple approach exhibits some room for improvements. However, it is worth notice that required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab. Also, thanks to the proposed approach, we have been able to outperform the baseline provided by the task organizers.

Acknowledgments

We extend our gratitude to the anonymous reviewers for their insightful comments and valuable suggestions, which have significantly enhanced the clarity and presentation of this paper.

References

Fabrice Colas and Pavel Brazdil. 2006. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Daniele Croce, Domenico Garlisi, and Marco Siino. 2022. An SVM ensemble approach to detect irony and stereotype spreaders on twitter. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2426–2432. CEUR-WS.org.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dominique Geissler, Dominik Bär, Nicolas Pröllochs, and Stefan Feuerriegel. 2023. Russian propaganda on social media during the 2022 invasion of ukraine. *EPJ Data Science*, 12(1).

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29,*

2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1746–1751. ACL.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A. Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Canadian AI*, volume 4013 of *Lecture Notes in Computer Science*, pages 395–406. Springer.

Joshua Littenberg-Tobias, G. R. Marvez, Garron Hillaire, and Justin Reich. 2022. Comparing few-shot learning with GPT-3 to traditional machine learning approaches for classifying teacher simulation responses. In *AIED (2)*, volume 13356 of *Lecture Notes in Computer Science*, pages 471–474. Springer.

Francesco Lomonaco, Gregor Donabauer, and Marco Siino. 2022. COURAGE at checkthat!-2022: Harmful tweet detection using graph neural networks and ELECTRA. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 573–583. CEUR-WS.org.

Francesco Lomonaco, Marco Siino, and Maurizio Tesconi. 2023. Text enrichment with japanese language to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2708–2716. CEUR-WS.org.

Beatriz Matias Santana Maia, Maria Clara Falcão Ribeiro de Assis, Leandro Muniz de Lima, Matheus Becali Rocha, Humberto Giuri Calente, Maria Luiza Armini Correa, Danielle Resende Camisasca, and Renato Antonio Krohling. 2024. Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer. *Expert Systems with Applications*, 241:122418.

Stefano Mangione, Marco Siino, and Giovanni Garbo. 2022. Improving irony and stereotype spreaders detection using data augmentation and convolutional neural network. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2585–2593. CEUR-WS.org.

Zong Meng, Zhaohui Zhang, Yang Guan, Jimeng Li, Lixiao Cao, Meng Zhu, Jingjing Fan, and Fengjie Fan. 2024. A hierarchical transformer-based adaptive metric and joint-learning network for few-shot rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(3).

Mohsen Miri, Mohammad Bagher Dowlatshahi, Amin Hashemi, Marjan Kuchaki Rafsanjani, Brij B Gupta, and W Alhalabi. 2022. Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*, 37(12):11319–11341.

Fuad Muftie and Muhammad Haris. 2023. Indobert based data augmentation for indonesian text classification. In *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, page 128 – 132.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Marco Siino. 2024a. Badrock at semeval-2024 task 8: Distilbert to detect multigenerator, multidomain and multilingual black-box machine-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024b. T5-medical at semeval-2024 task 2: Using t5 medical embeddings for natural language inference on clinical trial data. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024c. Transmistral at semeval-2024 task 10: Using mistral 7b for emotion discovery and reasoning its flip in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2021. Detection of hate speech spreaders using convolutional neural networks. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 2126–2136. CEUR-WS.org.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2022a. Fake news spreaders detection: Sometimes attention is not all you need. *Information*, 13(9):426.

Marco Siino, Marco La Cascia, and Ilenia Tinnirello. 2022b. Mcrock at semeval-2022 task 4: Patronizing and condescending language detection using multi-channel cnn, hybrid lstm, distilbert and xlnet. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 409–417. Association for Computational Linguistics.

Marco Siino, Francesco Lomonaco, and Paolo Rosso. 2024a. Backtranslate what you are saying and i will tell who you are. *Expert Systems*, n/a(n/a):e13568.

Marco Siino, Maurizio Tesconi, and Ilenia Tinnirello. 2023. Profiling cryptocurrency influencers with few-shot learning using data augmentation and ELECTRA. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2772–2781. CEUR-WS.org.

Marco Siino and Ilenia Tinnirello. 2023. Xlnet with data augmentation to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2763–2771. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2022. T100: A modern classic ensemble to profile irony and stereotype spreaders. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2666–2674. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024b. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Felipe Bonow Soares, Anatoliy Gruzd, and Philip Mai. 2023. Falling for russian propaganda: Understanding the factors that contribute to belief in pro-kremlin disinformation on social media. *Social Media and Society*, 9(4).

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

José Medardo Tapia-Téllez and Hugo Jair Escalante. 2020. Data augmentation with transformers for text classification. In *Advances in Computational Intelligence*, pages 247–259, Cham. Springer International Publishing.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Dang Van Thin, Duong Ngoc Hao, and Ngan Luu-Thuy Nguyen. 2023. Vietnamese sentiment analysis: An overview and comparative study of fine-tuning pretrained language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Trans. Circuits Syst. Video Technol.*, 33(12):7789–7802.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Veysel Çakmak. 2023. *Social media use and propaganda techniques: An evaluation of the Ukraine-Russia war*. IGI Global.

Mashee at SemEval-2024 Task 8: The Impact of Samples Quality on the Performance of In-Context Learning for Machine Text Classification

Areeg Fahad Rasheed
College of Information Engineering
Al-Nahrain University
Baghdad, Iraq
areeg.fahad@coie-nahrain.edu.iq

M. Zarkoosh
Software Engineering
Computiq
Baghdad, Iraq
m94zarkoosh@gmail.com

Abstract

Within few-shot learning, in-context learning (ICL) has become a potential method for leveraging contextual information to improve model performance on small amounts of data or in resource-constrained environments where training models on large datasets is prohibitive. However, the quality of the selected sample in a few shots severely limits the usefulness of ICL. The primary goal of this paper is to enhance the performance of evaluation metrics for in-context learning by selecting high-quality samples in few-shot learning scenarios. We employ the chi-square test to identify high-quality samples and compare the results with those obtained using low-quality samples. Our findings demonstrate that utilizing high-quality samples leads to improved performance with respect to all evaluated metrics.

1 Introduction

The advent of large language models (LLMs) like GPT-3.5 has brought about transformative capabilities, seamlessly handling tasks like question answering, essay writing, and problem-solving (Aljanabi et al., 2023; Wu et al., 2023; Rasheed et al., 2023a). However, this technological advancement necessitates careful consideration of its associated challenges. Concerns regarding the potential impact on creativity and ethical implications, particularly concerning the generation of deepfakes (Tang et al., 2023), warrant careful attention (RAYMOND, 2023). Additionally, the limitations of LLMs, including the possibility of producing erroneous information, require rigorous evaluation and verification. The substantial energy consumption required for training LLMs on massive datasets raises environmental concerns, contributing to their carbon footprint. Moreover, plagiarism issues emerge as users may misuse the generated content, either inadvertently or intentionally (Hadi et al., 2023).

Various models have been introduced in recent years designed to distinguish text generated by humans from that created by machines(Mitchell et al., 2023). Examples include GPTZero(gpt), AI Content Detector(cop), and AI Content Detector by Writer (wri) among others. Some of these models are trained on specific datasets, while others are commercially available. Designing and implementing LLMs for classification tasks requires substantial resources and computational power, which are often only accessible to institutions and governments. Therefore, various optimization models, such as LoRA (Hu et al., 2021), distillation(Hsieh et al., 2023), quantization(Dettmers et al., 2022), and in-context learning (Liu et al., 2022), have been developed to reduce the resource requirements for LLM implementation. This paper focuses on In Context Learning (ICL) (Liu et al., 2022), which utilizes the capabilities of other models to enhance their ability to classify AI-generated text.

In Context Learning (ICL) is a Natural Language Processing (NLP) technique utilized to enable Large Language Models (LLMs) to learn new tasks based on minimal examples. This technique proves powerful in scenarios where training models on extensive datasets is impractical or when there are constraints on dataset availability for a specific task. ICL operates on the premise that humans can often acquire new tasks through analogy or by observing a few examples of task performance. It can be employed without any examples and is referred to as zero-shot learning. Alternatively, if the input includes one example, it is termed one-shot learning, and if it contains more than one, it is known as few-shot learning. This paper focuses on the application of few-shot learning within the context of ICL(Ahmed and Devanbu, 2022; Kang et al., 2023).

In this study, our focus lies exclusively on few-shot learning. We present a methodology that leverages the chi-square statistic (Rasheed et al., 2023b;

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 60–63

Lancaster and Seneta, 2005) to select samples for few-shot learning and evaluate its impact on the performance of a machine-generated text classification model. We work on task A English language only (Wang et al., 2024).

2 Dataset

The dataset employed for Task A comprises two main components. The first part, derived from human writing, was collected from diverse sources including WikiBidia, WikiHow, Reddit, ArXiv, and PeerRead. The second part consists of a machine-generated text produced by ChatGPT, Cohere, Dolly-v2, and BLOOMz(Muennighoff et al., 2023). For further details, please refer to the associated paper (Wang et al., 2023).

3 Chi-square

Chi-square is a statistical test used to assess the independence of two categorical variables. It calculates the difference between observed and expected frequencies of outcomes, and a larger chi-square value indicates a stronger rejection of independence. In text analysis, chi-square can be used to identify keywords that are more likely to occur in one category than another, making it useful for feature selection and text classification. We computed the chi-square values for each training sample and recorded the sample index with the highest and lowest chi-square values for both human-generated and machine-generated samples. Table I displays the index and corresponding chi-square values for each of these instances. We will use X^2 to refer to chi-square (Lancaster and Seneta, 2005).

Table 1: Indices and chi-square values for highest/lowest in human-generated and machine-generated text

Name	Index #	X^2 Value
Highest X^2 (Human)	70873	1351.59
Lowest X^2 (Human)	85726	1.21
Highest X^2 (Machine)	2426	1154.27
Lowest X^2 (Machine)	29111	0.8243

4 System overview

The system architecture is illustrated in Figure 1. The process starts with feeding the entire training dataset to a chi-square computation, where the chi-square value for each sample is calculated. Subsequently, the indices of the samples with the highest and lowest chi-square values are selected for both human-generated and machine-generated datasets using information from Table I. Next, context learning is prepared. Initially, multiple templates were tested, and the one presented in Figure 1 yielded the best results. This template is then fed with two samples: the first being the machine-generated sample with the highest chi-square value, and the second being the human-generated sample with the highest chi-square value. Due to context window size limitations, only the first 5000 characters of each sample are incorporated. This is applied to training samples exceeding 5000 characters to ensure the context learning size is not exceeded. Finally, the test sample is fed into the context-learning process. The Flan-T5 model large version is used. The results are then recorded and evaluated. The dev/test sample size was truncated to 3000. We also evaluated the system using samples with the lowest chi-square values and doing the same process.

5 Findings and Analysis

We employed the Flan-T5 Large model for both the development and testing datasets. We selected samples from both human-generated and machine-generated sources, with each sample limited to 5000 characters to avoid exceeding the token size limit. A total of four experiments were conducted. The first experiment utilized samples with high chi-square values from the development set. The second experiment focused on samples with the smallest chi-square values from the development set. The third experiment involved samples with high chi-square values from the test set. Finally, the fourth experiment utilized samples with low chi-square values from the test set. Table II presents all achieved results.

Based on the results presented in Table II, we can discuss several key points.

- The results highlight the crucial role of sample quality in the performance of in-context learning. By leveraging the chi-squared metric and prioritizing samples with high values, we essentially provide the Flan-T5 model with examples rich in diverse features. This choice enables the Flan-T5 model to learn more effectively, drawing substantial insights from the samples. Consequently, the model becomes more familiar with the provided data, ultimately enhancing its performance. In

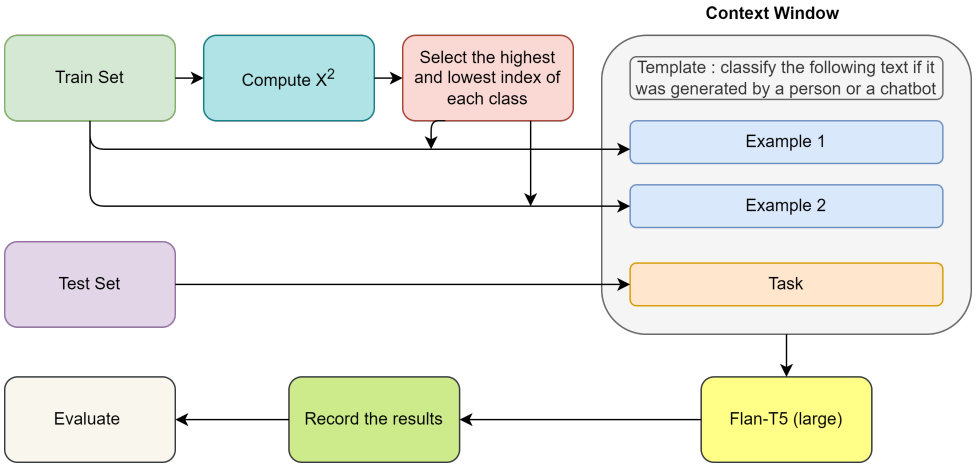


Figure 1: Proposed System Components

Dataset	Chi Type	Recall	Precision	F1-Score	Accuracy
Dev set	Lowest	46.92	46.90	46.84	46.92
	Highest	53.76	53.76	53.74	53.76
Test set	Lowest	55.04	55.07	55.03	55.27
	Highest	58.68	58.81	58.81	**55.99**

Table 2: Experiments results

contrast, selecting samples with lower quality leads to less optimal performance. This can be noticed for both the dev and test set. The main reason behind this is that words in the sample with high chi-square values contain the most distinctive features. This is because the chi-square test assigns high values to words that are frequent within a particular class but appear less frequently in other classes.Conversely, samples with lower chi-square values likely contain more random words that appear with similar frequency across all classes. In chi-square analysis, words that appear equally or approximately equally in each class receive lower scores.

- The classification of machine-generated text represents a novel frontier in machine learning, and the availability of datasets for this task is currently limited. The dataset used in this study was generated in 2023, marking it as a recent development and underscoring the lack of established benchmarks. Models that support in-context learning have not been trained extensively on such tasks, resulting in lower accuracy when applied. While examples with high-quality data can enhance model performance, it remain below the desired threshold. Hence, it is advisable to train the model directly on the dataset rather than relying on in-context learning.

- We have utilized the Flan-T5 model; however, other models can be employed to evaluate the performance of text classification machinery. We suggest considering alternatives such as bard, Jurassic-1 Jumbo, and ChatGPT.

6 Conclusion

This work presents a system for classifying human-generated and machine-generated text. The system leverages the combined strengths of in-context learning and Chi-square analysis. Chi-square is employed to select high-quality samples from the trainin dataset for few-shot learning in the in-context learning. We implement Flan-T5 model large version for in-context learning. Evaluation using accuracy, recall, precision, and F1-score demonstrates that selecting high-quality samples improves system performance for both dev and test. Furthermore, the results indicate that relying solely on in-context learning for new tasks like machine-generated text detection yields relatively low performance.

References

Ai content detector. Accessed on March 30, 2024.

Ai content detector by writer. Accessed on March 30, 2024.

Gptzero. Accessed on March 30, 2024.

Toufique Ahmed and Premkumar Devanbu. 2022. Few-shot training llms for project-specific code-summarization. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–5.

Mohammad Aljanabi, Mohanad Ghazi, Ahmed Hussein Ali, Saad Abas Abed, et al. 2023. Chatgpt: open possibilities. *Iraqi Journal For Computer Science and Mathematics*, 4(1):62–64.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Muhammad Usman Hadi, R Qureshi, A Shah, M Irfan, A Zafar, MB Shaikh, N Akhtar, J Wu, and S Mirjalili. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *TechRxiv*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Sungmin Kang, Juyeon Yoon, and Shin Yoo. 2023. Large language models are few-shot testers: Exploring llm-based general bug reproduction. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 2312–2323. IEEE.

Henry Oliver Lancaster and Eugene Seneta. 2005. Chi-square distribution. *Encyclopedia of biostatistics*, 2.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Areeg Fahad Rasheed, M Zarkoosh, Safa F Abbas, and Sana Sabah Al-Azzawi. 2023a. Arabic offensive language classification: Leveraging transformer, lstm, and svm. In *2023 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, pages 1–6. IEEE.

Areeg Fahad Rasheed, M Zarkoosh, and Sana Sabah Al-Azzawi. 2023b. The impact of feature selection on malware classification using chi-square and machine learning. In *2023 9th International Conference on Computer and Communication Engineering (ICCCE)*, pages 211–216. IEEE.

DANIEL RAYMOND. 2023. Disadvantages of large language models. Accessed on March 30, 2024.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. SemEval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136.

Puer at SemEval-2024 Task 4: Fine-tuning Pre-trained Language Models for Meme Persuasion Technique Detection

Jiaxu Dao, Zhuoying Li, Youbang Su, Wensheng Gong
School of Technology
Pu'er University
{daojiaxu, lizhuoying, suyoubang, gongwensheng}@peu.edu.cn

Abstract

The paper summarizes our research on multilingual detection of persuasion techniques in memes for the SemEval-2024 Task 4. Our work focused on English-Subtask 1, implemented based on a roberta-large pre-trained model provided by the transforms tool that was fine-tuned into a corpus of social media posts. Our method significantly outperforms the officially released baseline method, and ranked 7th in English-Subtask 1 for the test set. This paper also compares the performances of different deep learning model architectures, such as BERT, ALBERT, and XLM-RoBERTa, on multilingual detection of persuasion techniques in memes. The experimental source code covered in the paper will later be sourced from Github.

1 Introduction

Memes has been steadily increasing as human behavior as social media platforms have become more prevalent. This type of content is known for its rapid spread, achieved through the manipulation of audience psychology and the blurring of logical relationships.

Memes are generally made up of stacked images and text. The essence of its expression in order to generate an emotional effect is actually the skillful role of three persuasive strategies (Davison, 2012) in rhetorical portions:

1) Ethos: This involves the strategic employment of statements from individuals endowed with authority or credibility, thereby persuading the audience of the veracity of the content and augmenting its perceived legitimacy.

2) Pathos: By sharing personal anecdotes or experiences, memes forge a connection with the audience, evoking emotional resonance and deepening the affective engagement with the content.

3) Logos: The application of logical arguments and reasoning enhances the structural integrity and coherence of the message, fortifying its persuasiveness.

If we further split these three categories of persuasion strategies into twenty-two, scientists are able to obtain textual and visual features from memes for analysis. For instance, it is feasible to efficiently decrease or prevent the spread of hate speech, racial discrimination, and deceptive information by analysing memes, then simultaneously preserving the peace and stability of social media.

Memes can assist merchants in quickly capturing market trends, allowing them to carry out advertising and marketing operations more effectively and raise brand influence. Memes helps media workers in understanding the concerns of their audiences. Memes in politics have the potential to help voters demonstrate their policy views. The goal of the task is to classify corpora of text in memes and assign them to relevant persuasive strategies. Our work in SemEval-2024 Task 4 focuses on subtask 1,and this is a multi label classification task.

Our contrbutions can be highlighted as follows:

1) We explored new possibilities by screening models for news texts and multilingual corpus models. Fine-tuning using the social media posts corpus on the roberta-large model, and the experiment obtained hierarchical F1 of 0.647 on the English - Subtask 1 the dev set.

2) In SemEval-2024 Task 4, our model has an hierarchical F1 result of 0.66 in the English - Subtask 1 the test set, and our model ranks 7th on the leaderboard.

2 Related Work

Since the introduction of BERT in 2018(Devlin et al., 2018), its impact on the landscape of natural

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 64–69

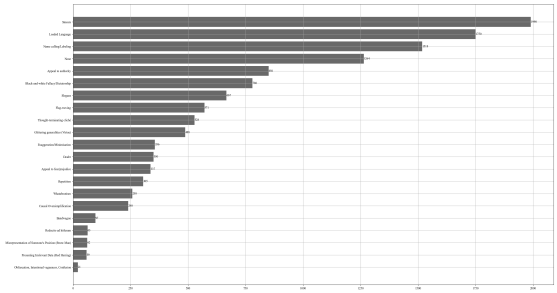


Figure 1: Number of Samples

ID	text	labels
67641	WHEN YOU'RE THE FBI, THEY LET YOU DO IT.	Thought-terminating cliché
66402	PUTIN'S SECRET CAMOUFLAGE ARMY	none
71251	Heaven has a Wall and strict immigration policies. Hell has open borders. President Donald J. Trump	Appeal to authority, Exaggeration/Minimisation
65282	ME VOTING ANTI-TRUMP IN 2016 ME VOTING ANTI-TRUMP IN 2020	Repetition

Table 1: Data Sample

language processing and multimodal analysis has been profound(Khare et al., 2021). BERT and its advanced derivatives, such as RoBERTa(Liu et al., 2019), XLM-RoBERTa(Xie et al., 2021), and ALBERT(Lan et al., 2019), have demonstrated their robust capabilities in a broad spectrum of applications, ranging from sentiment analysis to complex multimodal tasks that combine textual and visual data. Notably, RoBERTa has been recognized for its superior performance in accurately classifying sentiment(Liao et al., 2021), emotion(Kamath et al., 2022), and offensive content(Xu and Liu, 2023), highlighting the model's efficiency as a sophisticated text encoder.

The advent of these models has revolutionized the approach to analyzing diverse datasets and tasks, enabling nuanced understanding and processing of complex language patterns. This has been particularly evident in the domain of multimodal research, where BERT-based models have been instrumental in advancing the study of visual and textual data integration(Khan and Fu, 2021; He and Hu, 2021; Lee et al., 2021).

The success of these models in such a unique and culturally rich context exemplifies their broad applicability and the expanding frontiers of computational linguistics and content analysis. In conclusion, the inclusion of BERT and its variants in the analysis of persuasion techniques in memes marks a significant milestone in the field(Avvaru and Vobilisetty, 2020; Kougia and Pavlopoulos, 2021; Khedkar et al., 2022). It underscores the models' unparalleled flexibility and their emerging role in understanding the complexities of human communication in the digital age. As these models continue to evolve, their contribution to bridging the gap between textual and visual data analysis will undoubtedly pave the way for groundbreaking research and applications across diverse disciplines.

3 System Overview

3.1 Datasets

Our experiment employed four distinct datasets: the training set, validation set, development (dev) set, and test set, all formatted in JSON. The datasets feature a minimum sentence length of one. The training set comprises 7,000 entries, categorized into 20 distinct classes, showcasing an average sentence length of 19.94 and a maximum of 253. The validation set includes 500 entries, with an average sentence length of 18.85 and a maximum reaching 333. The development set, containing 1,000 samples, presents an average sentence length of 18.73 and a peak length of 145. Lastly, the test set encompasses 1,500 instances, with the sentences averaging 18 words in length.

Table 1 presents the sample dataset, illustrating the structured data used in our analysis.

Figure 1 sorts the distribution of categories in the training set in descending order of frequency, highlighting the frequency of each category. The term "None" denotes instances lacking specific classification. According to the depicted statistics, the category "Smear" constitutes the most significant portion of the dataset. In contrast, categories such as "Obfuscation", "Intentional vagueness" and "Confusion" represent the smallest proportions.

3.2 Pre-trained Model

The research team tends to choose from models related to news, tweets, and comments. The research team tested a number of models and deciding that Jochen Hartmann's sentiment-roberta-large-english-3-classes model (As shown in Table 2) while it received the best ratings. A comparison of outcomes from multiple models will be presented in the results section.

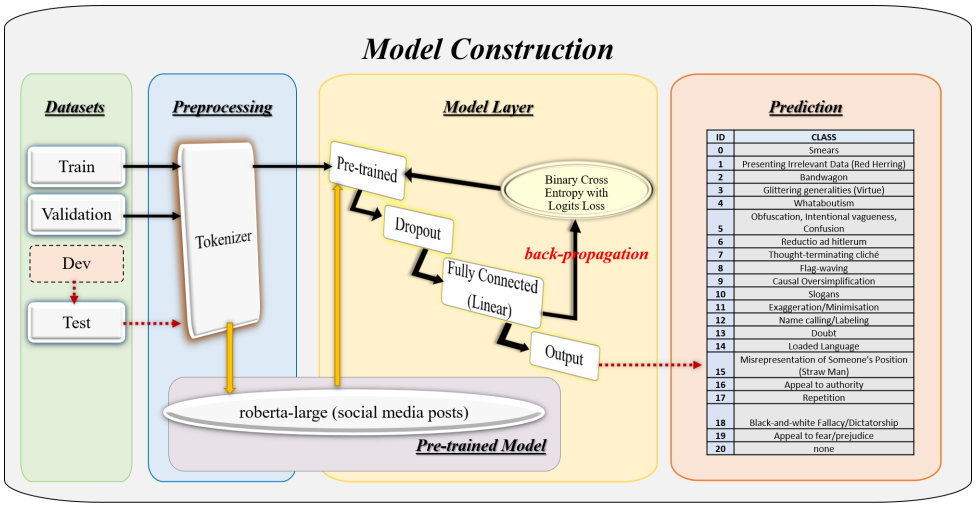


Figure 2: The architecture of model construction

ID	Model
1	bert-base-uncased
2	bert-base-multilingual-cased
3	albert-base
4	roberta-base
5	xlm-roberta-base
6	roberta-large
7	roberta-large(social media posts fine-tuned)

Table 2: Pre-trained Model

The sentiment-roberta-large-english-3-classes model (Hartmann et al., 2021) is trained based on tweets on social media platforms such as Twitter and Instagram, and includes text that is expected to include captions from the sender in the tweet image and comments from other observers. RoBERTa is used to construct the model. Achieving a hold out accuracy of 86.1 % , this model is used to evaluate user comments on posts and identify if the user is willing to buy a certain product. It demonstrates that the model has high robustness and a strong capacity to extract complicated text features.

3.3 Model Construction

In English-Subtask 1, to commence our experiment, we utilize the officially provided Train.json and Validation.json files as the training and validation datasets for supervised learning. Additionally, we assess subsequent results using the officially available dev dataset.

Secondly, we'll perform data preprocessing. The training and validation sets are fed into the Tokenizer, and the pre-trained model roberta-large(social media posts fine-tuned) is used for word segmentation and vectorization processing.

Following that, regarding model structure:

1) Input processing: Feed the pre-trained model with the processed token.

2) Dropout processing: Enter the dropout layer after model processing and set the inactivation probability to 0.1.

3) Linear fully connected layer: 1024 features are carried into the linear fully connected layer.

4) Loss function: For multi label classification jobs, Binary Cross Entropy With Logits Loss (Wang et al., 2022)serves as the loss function throughout the backpropagation gradient calculation procedure. BCEWithLogitsLoss comes with a sigmoid function that can convert predicted result values into probabilities, and can automatically handle numerical instability while preventing the sigmoid function from overflowing upwards or downwards (Yue et al., 2023).

Finally, the output layer is made up of 21 neurons, 20 of which are classified and one of which is none. The architecture of model construction is shown in Figure 2.

4 Experiment Setup

4.1 Evaluation Metrics

For English-Subtask 1, the participating systems are evaluated using standard evaluation metrics, including precision, recall, and hierarchical F1 scores.

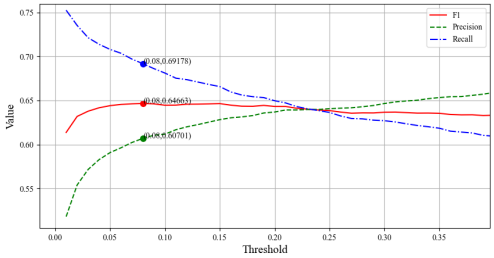


Figure 3: Impact of threshold on dev set

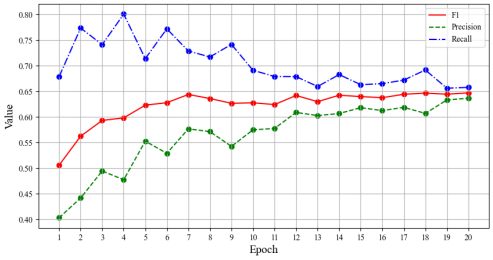


Figure 4: Impact of epoch on dev set

These metrics are calculated as follows:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (1)$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (2)$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (3)$$

The organizers provided baseline models for each subtask. For English - Subtask 1, the Hierarchical F1 scores for the baseline model were 0.358 on the development set and 0.369 on the test set.

4.2 Threshold Selection

The experimental results in training tasks will depend on the threshold selection. We select the most optimal hierarchical F1 value for determining the threshold, assuming that recall and precision are of identical significance. With a 0.01 interval, the experiment increased the threshold from 0 to 1.

The red dots on the hierarchical F1 value curve in Figure 3 represent the experimental results, which show that the most suitable threshold value for hierarchical F1 value is approximately 0.08. In our threshold parameter experimentation, we attained a recall rate of 0.69 and a precision of 0.60. Owing to the threshold being established at 0.08, Figure 3 incorporates merely a fraction of the experimental data. The hierarchical F1 scores start to decline as the threshold surpasses 0.4.

4.3 Epoch Selection

The epoch was raised in the experiment from 1 to 20 at intervals of 1. The Figure 4 illustrates that the Precision is low and unstable and the Recall value is high but swings continuously when the epoch is under seven. As a result of the Precision and hierarchical F1 values' continued continuous increase, the experimental model's instability will grow. The Recall steadily stabilizes as the epoch gets closer to 20, while the hierarchical F1 value also tends to stabilize.

In addition to the above parameters, other training parameters are set in Table 3 below.

Params	Value
num_train_epochs	20
per_device_train_batch_size	4
per_device_eval_batch_size	8
warmup_steps	500
weight_decay	0.01
logging_steps	100
save_strategy	epoch
evaluation_strategy	epoch
learning_rate	$1.5e^{-5}$
threshold	0.08

Table 3: Training Arguments

5 Results

As Table 4 shown, the model's performance on the development set revealed an hierarchical F1 score of 0.64, a precision of 0.63, and a recall of 0.65. The results indicate that our model achieves better results than other models. The performance of English-Subtask 1 on the test set yielded an hierarchical F1 score of 0.66, a precision of 0.65, and a recall of 0.67, ultimately securing the 7th position in the ranking.

6 Conclusion

In our participation in SemEval-2024 Task 4, specifically English-Subtask 1, we focused on addressing the challenge of multi-label text classification. Our study investigated the impact of various pre-trained models on experimental outcomes and the influence of different hyperparameters on

Model	F1	Precision	Recall
bert-base-uncased	0.59335	0.60017	0.58668
bert-base-multilingual-cased	0.58840	0.58235	0.59459
albert-base	0.59484	0.58081	0.60957
roberta-base	0.62268	0.60781	0.63829
xlm-roberta-base	0.58612	0.57927	0.59313
roberta-large	0.63679	0.61831	0.65640
roberta-large (social media posts fine-tuned)	0.64708	0.63666	0.65786

Table 4: Dev Set Results

model performance. Ultimately, the adoption of the roberta-large model fine-tuned on social media posts led to outstanding performance, achieving a hierarchical F1 score of 0.66 on the test set and securing a commendable 7th position among English-Subtask 1 participants.

In our experimentation, we did not pursue a finer-grained classification within the multi-label task. Moving forward, our future research direction will pivot towards fine-grained multi-label classification. This would entail optimizing the loss function or implementing multi-level classification techniques to enhance the model's generalization capabilities.

Acknowledgements

This work was supported by the 2024 Science and Technology special project of Pu 'er University(PYKJZX202401 Research on medical relationship extraction task based on pre-trained language model). The authors would like to thank the anonymous reviewers for their constructive comments.

References

Adithya Avvaru and Sanath Vobilisetty. 2020. Bert at semeval-2020 task 8: Using bert to analyse meme emotions. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1094–1099.

Patrick Davison. 2012. The language of internet memes. *The social media reader*, pages 120–134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jochen Hartmann, Mark Heitmann, Christina Schamp, and Oded Netzer. 2021. The power of brand selfies. *Journal of Marketing Research*, 58(6):1159–1177.

Jiaxuan He and Haifeng Hu. 2021. Mf-bert: Multimodal fusion in pre-trained bert for sentiment analysis. *IEEE Signal Processing Letters*, 29:454–458.

Rohan Kamath, Arpan Ghoshal, Sivaraman Eswaran, and Prasad Honnavalli. 2022. An enhanced context-based emotion detection model using roberta. In *2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6. IEEE.

Zaid Khan and Yun Fu. 2021. Exploiting bert for multimodal target sentiment classification through input space translation. In *Proceedings of the 29th ACM international conference on multimedia*, pages 3034–3042.

Yash Khare, Viraj Bagal, Minesh Mathew, Adithi Devi, U Deva Priyakumar, and CV Jawahar. 2021. Mmbert: Multimodal bert pretraining for improved medical vqa. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1033–1036. IEEE.

Sujata Khedkar, Priya Karsi, Devansh Ahuja, and Anshul Bahrani. 2022. Hateful memes, offensive or non-offensive! In *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 2*, pages 609–621. Springer.

Vasiliki Kougia and John Pavlopoulos. 2021. Multimodal or text? retrieval or bert? benchmarking classifiers for the shared task on hateful memes. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 220–225.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Sanghyun Lee, David K Han, and Hanseok Ko. 2021. Multimodal emotion recognition fusion analysis adapting bert with heterogeneous feature unification. *IEEE Access*, 9:94557–94572.

Wenxiong Liao, Bi Zeng, Xiuwen Yin, and Pengfei Wei. 2021. An improved aspect-category sentiment analysis model for text sentiment analysis based on roberta. *Applied Intelligence*, 51:3522–3533.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xing Wang, Wenxian Yang, Bo Qin, Kexiang Wei, Yunyu Ma, and Daibing Zhang. 2022. Intelligent monitoring of photovoltaic panels based on infrared detection. *Energy Reports*, 8:5005–5015.

Shuyi Xie, Jian Ma, Haiqin Yang, Lianxin Jiang, Yang Mo, and Jianping Shen. 2021. Pali at semeval-2021 task 2: fine-tune xlm-roberta for word in context disambiguation. *arXiv preprint arXiv:2104.10375*.

Meijia Xu and Shuxian Liu. 2023. Rb_bg_mha: A roberta-based model with bi-gru and multi-head attention for chinese offensive language detection in social media. *Applied Sciences*, 13(19):11000.

Xiaohan Yue, Danfeng Liu, Liguo Wang, Jón Atli Benediktsson, Linghong Meng, and Lei Deng. 2023. Iesrgan: Enhanced u-net structured generative adversarial network for remote sensing image super-resolution reconstruction. *Remote Sensing*, 15(14):3490.

Puer at SemEval-2024 Task2: A BioLinkBERT Approach to Biomedical Natural Language Inference

Jiaxu Dao, Zhuoying Li, Xiuzhong Tang, Xiaoli Lan, Junde Wang
School of Technology
Pu'er University
{daojiaxu, lizhuoying, tangxiuzhong, lanxiaoli, wangjunde}@peu.edu.cn

Abstract

This paper delineates our investigation into the application of BioLinkBERT for enhancing clinical trials, presented at SemEval-2024 Task 2. Centering on the medical biomedical NLI task, our approach utilized the BioLinkBERT-large model, refined with a pioneering mixed loss function that amalgamates contrastive learning and cross-entropy loss. This methodology demonstrably surpassed the established benchmark, securing an impressive F1 score of 0.72 and positioning our work prominently in the field. Additionally, we conducted a comparative analysis of various deep learning architectures, including BERT, ALBERT, and XLM-RoBERTa, within the context of medical text mining. The findings not only showcase our method's superior performance but also chart a course for future research in biomedical data processing. Our experiment source code is available on GitHub at: https://github.com/daojiaxu/semeval2024_task2.

1 Introduction

Clinical Trial Reports (CTRs) play a crucial role in documenting the methods and results of clinical trials(Jullien et al., 2023a; Vladika and Matthes, 2023). It contains a detailed overview of participant circumstances, intervention experiment descriptions, experimental results, and adverse events that happened in the participants. Natural Language Inference is a valuable technique for analyzing experimental data in CTR and interpreting the results. Natural Language Inference is able to analyze logical linkages, consistency, and contradictions in a document. It can assist detect logical relationships in text automatically, identify potential conflict areas fast, and improve decision-making accuracy and efficiency. Researchers can better gather and analyze clinical trial data by using Natural Language Inference techniques, which promotes medical quality improvement(Jullien et al., 2023b).

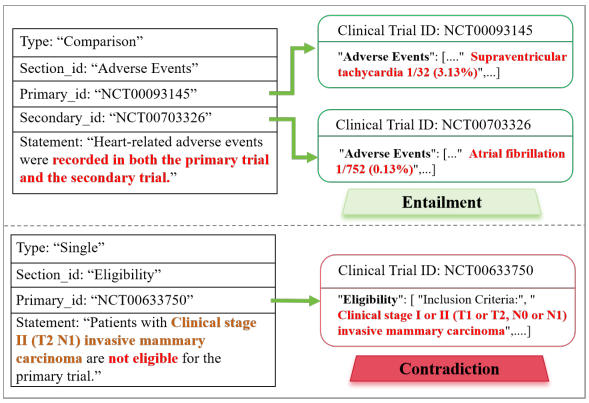


Figure 1: Dataset Example

The Figure 1 shows the example dataset used in this work. The dataset includes two forms of CTR: single and comparison. A single type CTR can retrieve relevant evidence using a Primary Id. To retrieve two relevant pieces of evidence using comparison type CTR, Primary Id and Secondary Id must be used simultaneously.

As an illustration, in the first instance, CTR represents "Heart-related adverse events were recorded in both the primary trial and the secondary trial." Searching for the matching components of the two pieces of evidence reveals that there are heart-related adverse effects, such as supraventricular tachycardia and atrial fibrosis. As a consequence, the first example is labeled as "Entailment"(Alsuhaibani, 2023). In a comparable way, in the second example, CTR believes that "Patients with clinical stage II (T2 N1) invasive breast cancer are not eligible for the primary trial." However, the participation conditions in the gathered evidence clearly show that individuals with clinical stage I or II (T1 or T2, N0 or N1) invasive mammary carcinoma match the criteria. As a result, the second case is labeled "Contradiction"(Liu et al., 2021; Zhou et al., 2023).

In the quest to push the frontiers of biomedical natural language understanding, SemEval-2024

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 70–75

Task 2 has emerged as a critical arena for testing the efficacy of AI models in parsing complex medical texts(Jullien et al., 2024). Engaging with this challenge, our work utilizes BioLinkBERT to set new benchmarks in the safety and accuracy of clinical trial inference(Ida et al., 2023; Karkera et al., 2023; Kanakarajan et al., 2022). This endeavor not only underscores the significance of developing robust NLI systems but also highlights our commitment to contributing meaningful innovations to the biomedical domain(Wang et al., 2023; Mahendra et al., 2023; Pahwa and Pahwa, 2023). Through this paper, we aim to share our methodologies, findings, and the implications they hold for the broader field of medical research, hoping to inspire further advancements and collaborative efforts in this vital area of study.

We created a number of attempts using the above dataset, and the following additions were contributed to our work:

1) We have designed a new loss function by combining the ideas of cross entropy and contrastive learning. This loss function can flexibly adjust parameters according to actual situations and has strong adaptability.

2) We have performed fine-tuning on the BioLinkBERT-large model and finally ranked 15th, achieving an F1 score of 0.72, a score of 0.59 in Faithfulness, and a score of 0.64 in Consistency.

2 System Description

2.1 Data Preprocessing

For this experiment, the training dataset was segmented into four distinct categories: Statement, Section, First Evidence, and Second Evidence. To facilitate precise identification of these text segments by the BioLinkBERT-large model, we employed the token "[SEP]" as a delineator for segment segmentation. This approach ensured that the model could accurately recognize and process the varied input text paragraphs, thereby enhancing its ability to understand and interpret the context and relationships within the data. This method of data preparation was crucial in optimizing the model's performance by providing clear structural demarcations within the training set.

More precisely, we create each input sample as shown in Figure 2.

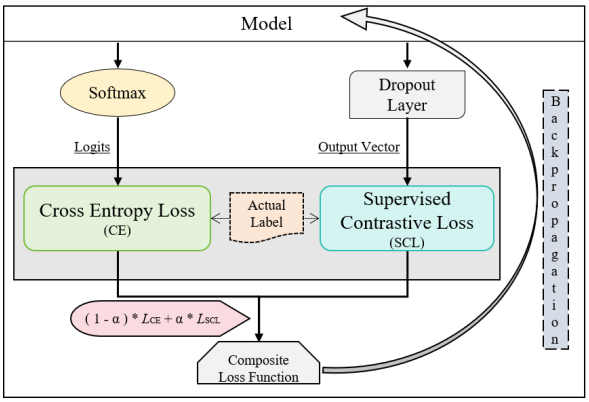


Figure 3: Composite Loss Function

Figure 2: The Architecture of Tokenizer

2.2 Model Construction

BioLinkBERT-large Model. In the domain of biological medicine, the BioLinkBERT model has been shown to be superior to the BERT model due to its ability to learn information across documents (Yasunaga et al., 2022). BioLinkBERT outperformed other models (BERT, BioMegatron, PubMedBERT, BioClinicalBERT, BioMedLM, BioGPT) in extracting the association between microorganisms and diseases from biomedical literature, with F1 precision and recall more than 0.8 (Karkera et al., 2023). The optimal accuracy was obtained in the histopathology image captioning challenge by integrating the BioLinkBERT target model with the image feature extractor ConvNexT Large (Elbedwehy et al., 2023). When compared to PubMedBERT and ChatGPT, the BioLinkBERT has demonstrated superior performance in all aspects in benchmark trials focused on biomedical text production and mining(Chen et al., 2023). The model we use is based on the BioLinkBERT large model that has been fine tuned from the MNLI and SNLI datasets.

Design of Loss Function. In the training phase, our loss function is bifurcated into two pivotal components. The initial segment utilizes the cross entropy loss function (`CrossEntropyLoss()`)(Zhang and Sabuncu, 2018), which first computes the predicted probability values via a softmax function. Subsequently, it leverages the cross entropy loss to quantify the deviation between these predicted probabilities and the actual labels, a process encapsulated by the symbol CE. The latter segment incorporates the supervised contrastive learning

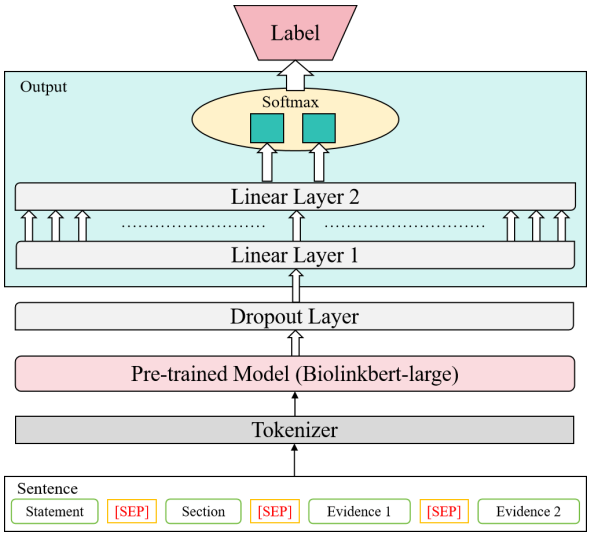


Figure 4: The Structure of System

loss function (`SupConLoss()`)(Khosla et al., 2020). Here, vectors generated post-processing by the pre-trained model are juxtaposed against the true labels to ascertain the contrastive learning loss, denoted as SCL.

Simultaneously, we have instituted a threshold parameter α to modulate the significance of each loss component. By amalgamating CE and SCL in accordance with this threshold, we obtain the composite loss. This loss is then subjected to back-propagation to minimize its magnitude, thereby aligning the predicted values more closely with the actual values. This methodology underscores our strategic approach to loss optimization, blending traditional and contrastive learning mechanisms to enhance model accuracy and performance.

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \cdot \log P \tag{1}$$

$$L_{SCL} = \sum_{i=1}^{N} \frac{-1}{N_{y_i} - 1} \sum_{\substack{j=1 \\ j \neq i}}^{N} 1_{y_i = y_j} \cdot \log \left(\frac{\exp(\Phi(x_i) \cdot \Phi(x_j)/\tau)}{\sum_{\substack{k=1 \\ k \neq i}}^{N} \exp(\Phi(x_i) \cdot \Phi(x_k)/\tau)} \right) \tag{2}$$

$$Loss = (1 - \alpha) * L_{CE} + \alpha * L_{SCL} \tag{3}$$

The Supervised Contrastive Learning (SCL) loss, as delineated in Equation (2), plays a pivotal role in the model's learning process by promoting the aggregation of examples from the same class while concurrently driving apart examples from distinct classes. Within a given batch, examples are meticulously grouped based on their corresponding labels, ensuring that the learning process is finely attuned to the nuances of class similarity and diversity. This is achieved through the implementation of the indicator function $1_{y_i=y_j}$, which is designed to ensure that the loss calculation exclusively considers pairs of examples (i, j) that, while sharing the same label, are distinct entities $(i \neq j)$. This deliberate focus on fostering intra-class cohesion and inter-class distinction is fundamental to augmenting the model's discriminative capabilities. A critical aspect of this approach is the use of N_{y_i}, which denotes the count of examples within the batch that share the same label as example i. This count is instrumental in normalizing the contribution of positive pairs to the loss, thereby ensuring that the SCL loss effectively enhances the model's proficiency in distinguishing between classes. This proficiency is further reinforced by the SCL loss's capacity to adjust based on the relative distances of examples within the embedding space, taking into account both positive pairs (belonging to the same class) and negative pairs (belonging to different classes), with N_{y_i} playing a crucial role in normalizing these effects based on the representation of each class within the batch.

This design strategy excels in leveraging annotated data to its fullest potential, significantly enhancing the model's generalization capabilities and the discriminative power of its feature representations. The cross-entropy loss function plays a pivotal role in assessing model performance by quantifying the discrepancy between predicted outputs and actual labels. Concurrently, the supervised contrastive learning loss function is instrumental in refining the discriminative capacity of feature representations, thereby bolstering classification accuracy. This dual-faceted approach not only ensures a comprehensive evaluation of model quality but also fosters a more nuanced understanding and representation of data features, which is crucial for achieving high precision in predictive tasks.

Model Layer Description. The levels in our model are as follows:

1) Sentence Input Layer: The model feeds the tokenizer with the text that was described in 2.1 as the training set.

2) Pre-trained Model Layer: To process the to-

Model	Loss	F1	Precision	Recall	Faithfulness	Consistency
bert-base-uncased	ce	0.6556	**0.956**	0.4989	0.0335	0.396
	ce+scl	0.6474	0.944	0.4926	0.0486	0.3931
albert-base	ce	0.6127	0.788	0.5012	0.1805	0.44
	ce+scl	0.6447	0.784	0.5474	0.2361	0.4951
biolinkbert-large	ce	0.7042	0.824	0.6149	0.4629	0.5971
	ce+scl	**0.7166**	0.764	**0.6749**	**0.5914**	**0.638**

Table 1: Comparative results of experiments in the test set

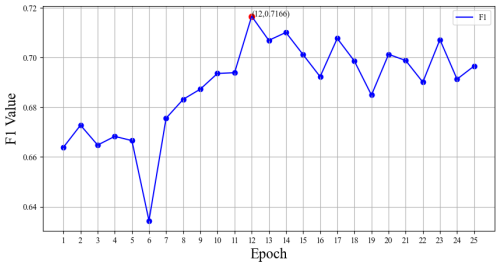


Figure 5: F1 Changes at Different Epochs on The Test Set

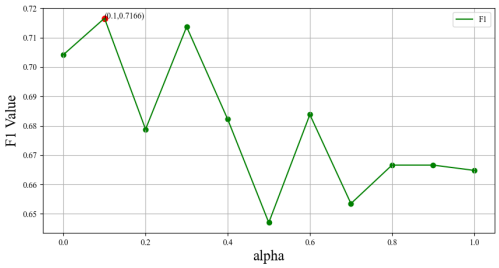


Figure 6: F1 Changes at Different Alpha on The Test Set

kenized text and produce the resultant vector representation of the text, the model makes use of the pre-trained model Biolinkbert-large.

3) Dropout layer: We implemented inactivation rate of 0.2 on the result vector to promote robustness and prevent overfitting of the model.

4) Linear Layer: To further process and transform vectors, the model employs two linear layers in the output module.

5) Softmax Function: Lastly, the model transforms the linear layer's output into a probability distribution by using the softmax function.

The loss function shown in Figure 3 was employed for backpropagation during the model training phase.The model's accuracy and real performance can be enhanced by adjusting the loss function parameter Alpha based on the current situation.

2.3 Hyper-parameters Fine-tuning

Epoch Selection. To ascertain the optimal F1 score, our experiment methodically adjusted the training duration, varying the epoch count from 1 to 20 in increments of one. At each epoch, we meticulously documented the corresponding F1 scores. As depicted by the blue line in Figure 5, a detailed analysis reveals that the F1 score peaks at epoch 12. This finding underscores the significance of epoch selection in maximizing model performance, illustrating that a carefully calibrated training period can significantly influence the effectiveness of the model's predictive accuracy.

Alpha Setting. Building upon this groundwork, we embarked on a series of experiments aimed at identifying the optimal value of alpha within the loss function, meticulously adjusting alpha from 0.1 to 1 in increments of 0.1. This systematic variation is represented by the green line in the accompanying graph. Through careful analysis, the ideal F1 score was observed when alpha was set to 0.1. This discovery not only highlights the critical role of alpha in tuning the loss function for enhanced model performance but also establishes a direct correlation between the fine-tuning of alpha and the achievement of peak predictive precision.

3 Experimental Results

In our methodology, we conducted two control trials by varying the loss function parameter Alpha, and selected three models(BERT-base-uncased(Devlin et al., 2018), ALBERT-base(Lan et al., 2019), and Biolinkbert-large)as outlined in

Table 1, aligning with the structure of our experiment. Subsequent to a rigorous examination of the experimental outcomes, it became evident that the experimental cohort employing the composite CE+SCL loss function surpassed the cohort utilizing the standalone CE loss function. This enhancement was observed across multiple metrics, including F1 score, recall, faithfulness, and consistency, specifically within the ALBERT-base and Biolinkbert-large models.

Upon comprehensive evaluation, the Biolinkbert-large model consistently demonstrates outstanding stability and superior performance. While the BERT-based-uncased model, employing the Cross-Entropy (CE) loss function, achieved the highest Precision score, it also registered relatively lower scores in terms of Faithfulness and Consistency. To encapsulate, the Biolinkbert-large model has exhibited exceptional proficiency in addressing this particular challenge.

4 Conclusion

This study has presented a comprehensive analysis of the effectiveness of BioLinkBERT in enhancing clinical trials. Our research has meticulously fine-tuned the BioLinkBERT-large model with a novel mixed loss function. The experimental results, particularly the achievement of an F1 score of 0.72, underscore the potential of leveraging advanced pre-trained language models in medical research. Our findings suggest that the integration of contrastive learning and cross-entropy loss functions significantly improves the model's performance, indicating a promising direction for future research in biomedical text mining.

Moreover, the success of this project opens new avenues for exploring the application of language models like BioLinkBERT in other domains of healthcare and medical research. Future work could focus on expanding the dataset, experimenting with different architectures, and exploring the impact of domain-specific adaptations on model performance. This could potentially lead to breakthroughs in how we process, understand, and derive insights from clinical trial reports, ultimately contributing to the advancement of medical science and patient care.

Acknowledgements

This work is supported by the 2024 Science and Technology special project of Pu'er University (PYKJZX202401 Research on medical relationship extraction task based on pre-trained language model). The authors would like to thank the anonymous reviewers for their insightful feedback.

References

Mohammed Alsuhaibani. 2023. Deep learning-based sentence embeddings using bert for textual entailment. *International Journal of Advanced Computer Science and Applications*, 14(8).

Qijie Chen, Haotong Sun, Haoyang Liu, Yinghui Jiang, Ting Ran, Xurui Jin, Xianglu Xiao, Zhimin Lin, Hongming Chen, and Zhangmin Niu. 2023. An extensive benchmark study on biomedical text generation and mining with chatgpt. *Bioinformatics*, 39(9):btad557.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Samar Elbedwehy, T Medhat, Taher Hamza, and Mohammed F Alrahmawy. 2023. Enhanced descriptive captioning model for histopathological patches. *Multimedia Tools and Applications*, pages 1–20.

Ryuki Ida, Makoto Miwa, and Yutaka Sasaki. 2023. Biomedical document classification with literature graph representations of bibliographies and entities. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 385–395.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Donal Landers, and André Freitas. 2023a. Nli4ct: Multi-evidence natural language inference for clinical trial reports. *arXiv preprint arXiv:2305.03598*.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Donal Landers, and André Freitas. 2023b. Semeval-2023 task 7: Multi-evidence natural language inference for clinical trial data. *arXiv preprint arXiv:2305.02993*.

Kamal Raj Kanakarajan, Bhuvana Kundumani, Abhijith Abraham, and Malaikannan Sankarasubbu. 2022. Biosimcse: Biomedical sentence embeddings using contrastive learning. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 81–86.

Nikitha Karkera, Sathwik Acharya, and Sucheendra K Palaniappan. 2023. Leveraging pre-trained language

models for mining microbiome-disease relationships. *BMC bioinformatics*, 24(1):290.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Ruishan Liu, Shemra Rizzo, Samuel Whipple, Navdeep Pal, Arturo Lopez Pineda, Michael Lu, Brandon Arnieri, Ying Lu, William Capra, Ryan Copping, et al. 2021. Evaluating eligibility criteria of oncology trials using real-world data and ai. *Nature*, 592(7855):629–633.

Rahmad Mahendra, Damiano Spina, and Karin Verspoor. 2023. Ittc at semeval 2023-task 7: Document retrieval and sentence similarity for evidence retrieval in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation, Toronto, Canada. Association for Computational Linguistics*.

Bhavish Pahwa and Bhavika Pahwa. 2023. Bphigh at semeval-2023 task 7: Can fine-tuned cross-encoders outperform gpt-3.5 in nli tasks on clinical trial data? In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1936–1944.

Juraj Vladika and Florian Matthes. 2023. Sebis at semeval-2023 task 7: A joint system for natural language inference and evidence retrieval from clinical trial reports. *arXiv preprint arXiv:2304.13180*.

Weiqi Wang, Baixuan Xu, Tianqing Fang, Lirong Zhang, and Yangqiu Song. 2023. Knowcomp at semeval-2023 task 7: Fine-tuning pre-trained language models for clinical trial entailment identification. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1–9.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. Thifly research at semeval-2023 task 7: A multi-granularity system for ctr-based textual entailment and evidence retrieval. *arXiv preprint arXiv:2306.01245*.

NRK at SemEval-2024 Task 1: Semantic Textual Relatedness through Domain Adaptation and Ensemble Learning on BERT-based models

Nguyen Tuan Kiet[1,2] and **Dang Van Thin**[1,2]
[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
21521042@gm.uit.edu.vn, thindv@uit.edu.vn

Abstract

This paper describes the system of the team NRK for Task A in the SemEval-2024 Task 1: Semantic Textual Relatedness (STR). We focus on exploring the performance of ensemble architectures based on the voting technique and different pre-trained transformer-based language models, including the multilingual and monolingual BERTology models. The experimental results show that our system has achieved competitive performance in some languages in Track A: Supervised, where our submissions rank in the Top 3 and Top 4 for Algerian Arabic and Amharic languages. Our source code is released on the GitHub site[1].

1 Introduction

The SemEval-2024 Task 1 (Ousidhoum et al., 2024b) aims at detecting the degree of semantic relatedness between pairs of sentences across 14 different languages, encompassing Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu. This shared task has three main tasks, each focusing on different aspects of predicting semantic textual relatedness within sentence pairs.

Semantic Textual Relatedness (STR) is a task in Natural Language Processing (NLP) that aims to measure the degree of semantic relatedness between two text passages, typically sentences. STR plays a crucial role in various NLP applications, as it allows computers to understand the relationships between different pieces of text. As mentioned in (Abdalla et al., 2023), it is also employed in chatbots and dialogue systems to understand the user's intent and in question-answering systems to identify answer passages that are semantically related to the question. Additionally, STR finds applications in text summarization, where it helps identify the most important and semantically relevant sentences to create a concise summary of a longer document. STR also plays a role in text generation tasks, such as machine translation and dialogue systems, by guiding the model to generate text that is semantically related to the input or context. However, accurately measuring STR presents several challenges. One key challenge lies in capturing the nuances of language, such as synonyms, paraphrases, and ambiguity. Another challenge is dealing with different languages and cultural contexts, where semantic relationships might not be directly translatable.

Our team only focuses on addressing Track A in the shared task. Our approach is based on the domain adaption for different transformer-based models, and then we continue to fine-tune the pre-trained transformer-based models on the task-specific training data. Therefore, our system is able to leverage domain-specific knowledge to improve performance. Subsequently, we train a cross-encoder model on the adapted transformer-based models, harnessing its ability to capture semantic relatedness between sentence pairs effectively. To further enhance the robustness and performance of our predictions, we adopt a weighted voting technique to combine the outputs of multiple models.

2 Background

2.1 Problem Description

This study investigates the task of predicting Semantic Textual Relatedness (STR) between sentence pairs across 14 languages. Each sentence pair will be associated with a human-annotated relatedness score ranging from 0 (completely unrelated) to 1 (maximally related). There are three Tracks for participants, however, in our work, we only focus on Track A: The first task entails a supervised approach, wherein participants are tasked with developing systems that leverage labelled training

[1]https://github.com/KiRzEa/Semeval2024-SemanticTextualRelatedness

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 76–81

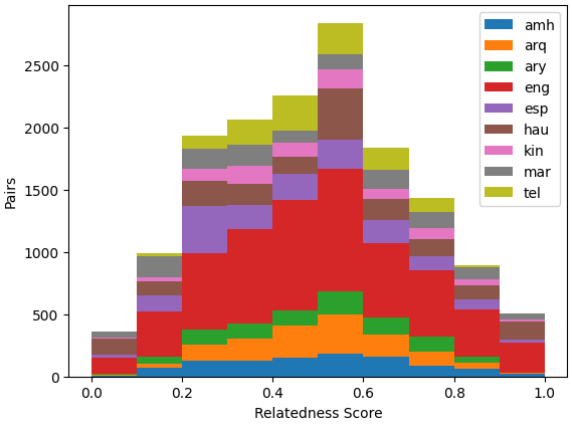


Figure 1: Relatedness Score distribution over languages on the training set.

datasets to infer the degree of semantic relatedness between sentence pairs.

2.2 Data Description

The dataset (Ousidhoum et al., 2024a) typically contains pairs of text along with their corresponding relatedness score, which indicates how semantically related the two fragments are.

Figure 1 shows the distribution of relatedness score over languages. Among the languages included in the dataset, English comprises the largest subset of sentence pairs. The remaining languages also contribute sentence pairs, albeit with varying degrees of representation. It is notable that while most languages exhibit relatedness score distributions spanning the entire range of 0 to 1, some languages demonstrate more limited distributions.

3 Related Work

STR is a fundamental concept which has been considered as an important role in language understanding tasks. Historically, many previous studies focused on semantic similarity, which aims to measure the likeness or resemblance between linguistic elements based on their meaning (Abdalla et al., 2023). Unlike semantic similarity, which often involves assessing the degree of overlap or similarity in meaning between words or phrases, STR involves determining the overall relatedness or closeness in meaning between pairs of sentences or longer textual units (Mohammad and Hirst, 2012). (Gabrilovich et al., 2007) proposed a novel method called Explicit Semantic Analysis (ESA) for fine-grained semantic representation of unrestricted natural language texts. The effectiveness of ESA is evaluated by automatically computing the degree of semantic relatedness between fragments of natural language text. Hussain et al. (2023) proposed a novel vector space model for computing semantic similarity and relatedness between concepts by aggregating taxonomic features from WordNet and Wikipedia.

With the emergence of deep learning models, Gu et al. (2023) introduced a novel Siamese Manhattan LSTM-SNP approach (SiMaLSTM-SNP) which combines Word2Vec and a 10-layer Attention strategy to represent and extract sentence pairs. The multi-head self-attention layer identifies text associations and redistributes hidden state weights. The last hidden state is extracted, and the relatedness score is calculated using the Manhattan distance. Hany et al. (2023) employed a two-layered approach. Firstly, embedding similarity techniques were utilized, leveraging seven different transformers to obtain vectors for each pair of sentences. Secondly, a classical machine learning regressor was trained on these seven vectors. This research highlights the potential of combining embedding similarity techniques with machine learning methods to enhance relatedness score assessment and other NLP tasks.

4 System Description

4.1 Approach

The diagram in Figure 2 illustrates our ensemble approach for Task A. The framework consists of two main layers: a layer of cross-encoder model, and a voting ensemble layer. Firstly, the input sentence pair is passed through a single encoder to produce a joint representation which captures the semantic relationship between the two sentences in the pair and produces a number ranging from 0 to 1. Following this, the predictions of chosen models are combined using the weighted voting technique with each weight determined by its performance in the development phase.

Our approach commences with domain adaptation on masked language modeling (MLM) task (3) which has been shown a powerful training strategy for learning sentence embeddings (Gururangan et al., 2020). To achieve this, we leverage each sentence in the sentence pairs of the training dataset to train MLM which is called In-domain corpus in Figure 2. This process involves masking certain tokens within the input sentences and training the model to predict the masked tokens based

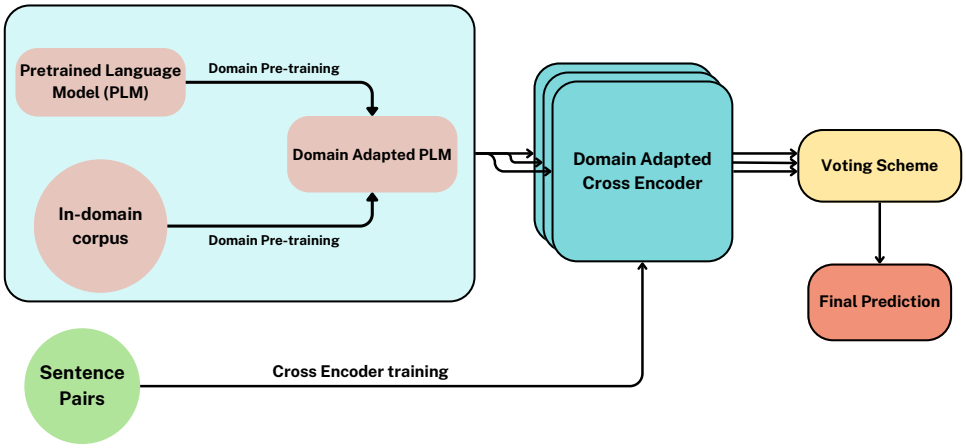


Figure 2: The overall framework of our system for the Track A: Supervised in the Semantic Textual Relatedness shared task.

on their context. In the next stage, we employ a cross-encoder architecture from Sentence-BERT (Reimers and Gurevych, 2019) which is a variant of the BERT model specifically designed for generating fixed-size sentence embeddings that capture semantic similarity between sentences. The cross-encoder architecture of SBERT processes sentence pairs jointly, encoding them into dense fixed-size vectors while considering their contextual information and semantic relationships. After obtaining the logits, we apply the sigmoid function to transform the logits into scores ranging from 0 to 1.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

This transformation ensures that the output scores are normalized and represent the degree of semantic relatedness between sentence pairs. To optimize the model during training, we utilize Binary CrossEntropy loss function $\mathcal{L}$ as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \quad (2)$$

Fine-tuning Language Model: As can be seen in Figure 2, we utilize the power of pre-trained contextual language models, encompassing BERT-based models which are BERT (**?**), DeBERTa-V3 (He et al., 2022), XLM-RoBERTa (Conneau et al., 2019) and E5 (Wang et al., 2022). To fine-tune the language models, we followed

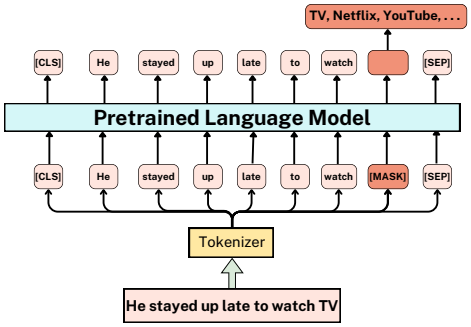


Figure 3: Masked language modelling task illustration for BERT-based models.

the approach of (Devlin et al., 2019), which is presented in detail below.

Voting Scheme: Our motivation for applying an ensemble approach is to take advantage of the performances of various models. Given predictions $\{\hat{y}_{\theta_1}, \hat{y}_{\theta_2}, .., \hat{y}_{\theta_n}\}$ of the n base regressors. We applied the weighted voting technique to merge the predictions of the base models. In our case, the individual regressors are treated based on their performance in the evaluation phase. We compute the weighted sum of the output of n regressors as the final prediction.

4.2 Pre-trained Contextual Language Models

We briefly explain the pre-trained language models used in this paper.

- **mBERT**: we use the multilingual version of BERT (Devlin et al., 2019) which is trained

Table 1: Results of our best submission compared with two top systems on 9 languages for Track A.

Track A1: Algerian Arabic		**Track A2: Amharic**		**Track A3: English**	
Team	**Score**	**Team**	**Score**	**Team**	**Score**
Top 1	0.6823	Top 1	0.8886	Top 1	0.8596
Top 2	0.6788	Top 2	0.8878	Top 3	0.8532
Ours (Top 3)	**0.6736**	**Ours (Top 4)**	**0.8641**	**Ours (Top 14)**	**0.8352**

Track A4: Hausa		**Track A5: Kinyarwanda**		**Track A6: Marathi**	
Team	**Score**	**Team**	**Score**	**Team**	**Score**
Top 1	0.7642	Top 1	0.8169	Top 1	0.9108
Top 2	0.7472	Top 2	0.8134	Top 2	0.8968
Ours (Top 8)	**0.6719**	**Ours (Top 6)**	**0.7568**	**Ours (Top 6)**	**0.8792**

Track A7:Moroccan Arabic		**Track A8: Spanish**		**Track A9: Telugu**	
Team	**Score**	**Team**	**Score**	**Team**	**Score**
Top 1	0.8625	Top 1	0.7403	Top 1	0.8733
Top 2	0.8596	Top 2	0.7310	Top 2	0.8643
Ours (Top 6)	**0.8269**	**Ours (Top 12)**	**0.6898**	**Ours (Top 8)**	**0.8341**

on the top 104 languages with the largest Wikipedia using a masked language modelling (MLM) objective with case sensitivity.

- **XLM-R**: XLM-R (Conneau et al., 2020) is another multilingual language model. It is pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages.
- **mDeBERTa-V3**: a DeBERTa (He et al., 2020) version improved the efficiency of original DeBERTa using ELECTRA-Style pre-training with Gradient Disentangled Embedding Sharing (He et al., 2022). In our case, we choose the multilingual version of DeBERTa-V3 which was pre-trained only on the ConmmonCrawl dataset and other versions, which are fine-tuned on the XNLI dataset and multilingual-NLI-26lang-2mil7 dataset (Laurer et al., 2024), respectively.
- **E5**: E5 (Wang et al., 2022) is trained in a contrastive manner with weak supervision signals from our curated large-scale text pair dataset. We chose monolingual (which is trained only in English) and multilingual versions for our task.

5 Experimental Setup

Data and Pre-processing: We utilized the official training set for training models. The development set was used to determine the weights for each model chosen to apply the voting technique based on their performance.

Configuration Settings: We implemented our models using the Trainer API from the Hugging Face library (Wolf et al., 2020) for the MLM task and employed the Cross Encoder architecture from SBERT (Reimers and Gurevych, 2019) for the Cross Encoder task.

- **MLM Task**: The maximum input length is set to 512 tokens, and the number of epochs is set to 10 with a batch size of 16 for all languages. During the training phase of the MLM, we set the MLM probability to 0.15, which means a token will be replaced with the [MASK] token in the input sequence with a probability of 0.15.
- **Cross Encoder Task**: The maximum input length is set to 512 tokens, and the number of epochs is set to 10 with a batch size of 16 for all languages.

We used the AdamW optimizer with a linear schedule warm-up technique for both the MLM task and the Cross Encoder task.

Submission Systems: We submitted the performance of the ensemble weighted voting model for all languages for both the development phase and evaluation phase and as mentioned above, the weights of each model based on its performance in the development phase and determined manually.

6 Results and Discussion

In this section, we present the official results of our final submission model for Track A in the SemEval

Table 2: Results of all the base models and our ensemble models on the development dataset.

Track A1: Algerian Arabic		Track A2: Amharic		Track A3: English	
Model	**Score**	**Model**	**Score**	**Model**	**Score**
XLMR-large	0.570	XLMR-large	0.878	XLMR-large	0.818
mBERT	0.566	mBERT	0.257	mBERT	0.798
mE5-base	0.559	mE5-base	0.828	mE5-base	0.805
mE5-large	0.523	mE5-large	0.889	mE5-large	0.824
mDeBERTa-v3-base	0.561	mDeBERTa-v3-base	0.859	mDeBERTa-v3-base	0.821
mDeBERTa-v3-xnli	**0.664**	mDeBERTa-v3-xnli	0.878	mDeBERTa-v3-xnli	0.823
-	-	-	-	E5-v2-large	0.828
Ensemble	0.659	**Ensemble**	**0.891**	**Ensemble**	**0.840**

Track A4: Hausa		Track A5: Kinyarwanda		Track A6: Marathi	
Model	**Score**	**Model**	**Score**	**Model**	**Score**
XLMR-large	0.785	XLMR-large	0.641	XLMR-large	0.858
mBERT	0.741	mBERT	0.651	mBERT	0.822
mE5-base	0.747	mE5-base	0.664	mE5-base	0.825
mE5-large	0.752	mE5-large	0.652	mE5-large	0.860
mDeBERTa-v3-base	0.718	mDeBERTa-v3-base	0.646	mDeBERTa-v3-base	0.829
mDeBERTa-v3-xnli	0.759	mDeBERTa-v3-xnli	0.662	mDeBERTa-v3-xnli	0.839
Ensemble	**0.791**	**Ensemble**	**0.665**	**Ensemble**	**0.862**

Track A7: Moroccan Arabic		Track A8: Spanish		Track A9: Telugu	
Model	**Score**	**Model**	**Score**	**Model**	**Score**
XLMR-large	0.833	XLMR-large	0.665	XLMR-large	0.803
mBERT	0.831	mBERT	0.673	mBERT	0.790
mE5-base	0.840	mE5-base	0.666	mE5-base	0.797
mE5-large	0.851	mE5-large	0.691	mE5-large	0.809
mDeBERTa-v3-base	0.816	mDeBERTa-v3-base	**0.729**	mDeBERTa-v3-base	0.805
mDeBERTa-v3-xnli	0.818	mDeBERTa-v3-xnli	0.701	mDeBERTa-v3-xnli	0.810
Ensemble	**0.860**	**Ensemble**	0.728	**Ensemble**	**0.827**

2024 Task 1, comparing them with the results of the two top-performing teams for each sub-track.

Table 1 showcases the performance of our ensemble model alongside that of the top two teams across nine tracks. Our system demonstrates competitive performance across four sub-tracks: Track A1 (Algerian Arabic), Track A2 (Amharic), Track A3 (English), and Track A7 (Moroccan Arabic). Additionally, we provide the results of both base models and ensemble systems on the development set. As indicated in Table 2, the ensemble gives better performance in most of the sub-tracks. Notably, we observe a decline in the performance of the ensemble on certain tracks (e.g., Track A1, Track A8) attributed to the presence of a base model that significantly outperforms the others and when this superior model is combined with the rest, it leads to a degradation in the overall performance of the ensemble that underscores the complexity of ensemble. In Track A2, the mBERT model was excluded from the ensemble due to its poor performance, the ensemble was thus formed using only the remaining models. Consequently, we opted for the ensemble model as the final submission system over the best model identified on the development set.

7 Conclusion

This paper introduces a straightforward yet effective ensemble architecture for Track A in the SemEval-2024 Task 1: Semantic Textual Relatedness. Our system leverages fine-tuning of pretrained transformer-based language models as base regressors, coupled with a weighted voting technique to amalgamate predictions from diverse base models. Experimental results demonstrate its competitive performance across select languages in Track A without any additional resources. For future works, we propose enhancing our system by integrating African transformer-based models and exploring data augmentation techniques to improve the overall performance.

Acknowledgements

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Evgeniy Gabrilovich, Shaul Markovitch, et al. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611.

Xu Gu, Xiaoliang Chen, Peng Lu, Xiang Lan, Xianyong Li, and Yajun Du. 2023. Simalstm-snp: novel semantic relatedness learning model preserving both siamese networks and membrane computing. *The Journal of Supercomputing*, pages 1–30.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Mena Hany, Mostafa Mohamed Saeed, Rana Reda Waly, Abdelrahman Ezzeldin Nagib, and Wael H Gomaa. 2023. Enhancing textual relatedness assessment with combined transformers-embedding similarity techniques and machine learning regressors. In *Proceeding of IMSA*, pages 13–18. IEEE.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention.

Muhammad Jawad Hussain, Heming Bai, Shahbaz Hassan Wasti, Guangjian Huang, and Yuncheng Jiang. 2023. Evaluating semantic similarity and relatedness between concepts by combining taxonomic and non-taxonomic semantic features of wordnet and wikipedia. *Information Sciences*, 625:673–699.

Moritz Laurer, Wouter Van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli. *Political Analysis*, 32(1):84–100.

Saif M Mohammad and Graeme Hirst. 2012. Distributional measures of semantic distance: A survey. *arXiv preprint arXiv:1203.1858*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, et al. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

BrainLlama at SemEval-2024 Task 6: Prompting Llama to detect hallucinations and related observable overgeneration mistakes

Marco Siino
Department of Electrical, Electronic
and Computer Engineering
University of Catania
Italy
marco.siino@unipa.it

Abstract

Participants in the SemEval-2024 Task 6 were tasked with executing binary classification aimed at discerning instances of fluent overgeneration hallucinations across two distinct setups: the model-aware and model-agnostic tracks. That is, participants must detect grammatically sound output which contains incorrect or unsupported semantic information, regardless of whether they had access to the model responsible for producing the output or not, within the model-aware and model-agnostic tracks. Two tracks were proposed for the task: a model-aware track, where organizers provided a checkpoint to a model publicly available on HuggingFace for every data point considered, and a model-agnostic track, where the organizers do not. In this paper, we discuss the application of a Llama model to address both the tracks. Our approach reaches an accuracy of 0.62 on the agnostic track and of 0.67 on the aware track.

1 Introduction

In the modern Natural Language Generated (NLG) domain, two interconnected challenges persist: neural models often produce linguistically fluent, yet inaccurate, output, while evaluation metrics primarily focus on fluency rather than accuracy. This situation leads to the phenomenon of "hallucinations," wherein neural networks generate output that sound plausible but deviate from the intended meaning, posing difficulties in automatic detection. However, in many NLG applications, the accuracy of output is paramount. For instance, generating translations that diverge from the source text undermines the effectiveness of machine translation systems. Also, as reported in recent survey papers, LLMs are prone to hallucinations, as proven in a variety of recent survey papers (Huang et al., 2023; Ji et al., 2023; Zhang et al., 2023). This LLMs drawback led to the proposal of SemEval-2024 Task 6 (Mickus et al., 2024), where participants were tasked with conducting detection of hallucinations across two subtracks: model-agnostic and model-aware. Put simply, participants were required to detect grammatically correct output containing incorrect or unsupported semantic information, regardless of access to the model responsible for generating them. In the literature, the task has been recently addressed with prompt engineering strategies that provide further context to the models to properly drive and control the models' output (Martino et al., 2023; Li et al., 2024).

To aid in this assignment, a dataset including references, inputs, checkpoints, and outputs from systems trained for three NLG tasks (definition, modeling, machine translation, and paraphrase generation) was provided. These systems were trained with varying levels of accuracy. The dataset included development and test sets annotated by a minimum of five annotators, with a majority vote establishing the gold label for binary annotations.

To address these objectives, there is an ongoing demand for automated tools capable of extracting and categorizing data, facilitating the classification of NLG content containing hallucinations. Recent advancements in machine and deep learning architectures have spurred heightened interest in Natural Language Processing (NLP). Substantial endeavors have been directed towards devising techniques for the automated identification and categorization of textual content accessible on the internet today. In the literature, to perform text classification tasks, several strategies have already been proposed (Kim, 2014; Siino et al., 2024a; Lomonaco et al., 2023).

To face with the task, we propose a Transformer-based approach which made use of Llama (Touvron et al., 2023). We used the model in a zero-shot setup described in the rest of this paper. Specifically, we prompted the latest pre-trained version of Llama with any sample in the dataset. Specifically, we provided a *context* and a *sentence*, asking the

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 82–87

model if the sentence was really supported by the context or was an example of hallucination.

The subsequent sections of the paper are structured as follows: Section 2 offers background information on Task 6, held at SemEval-2024. In Section 3, we outline the approach introduced in this study. Section 4 delves into the specifics of the experimental setup employed to reproduce our findings. The outcomes of the official task and relevant discussions are presented in Section 5. Finally, Section 6 concludes our study and suggests avenues for future research.

We make all the code publicly available and reusable on GitHub[1].

2 Background

This section furnishes background information regarding Task 6, held at SemEval-2024 (named, *SHROOM*). SHROOM participants are tasked with identifying grammatically correct output containing incorrect semantic information, regardless of their access to the model responsible for generating the output.

The data files are formatted as JSON lists, with each element representing a datapoint. Each datapoint corresponds to a different model production and includes the following details:

- Task (task): indicating the objective the model was optimized for.
- Source (src): the input provided to the models for the generation.
- Target (tgt): the intended reference "gold" text that the model should generate.
- Hypothesis (hyp): the actual output generated by the model.
- Annotator labels (labels): indicating whether each individual annotator considered this datapoint to be a hallucination or not.
- Majority-based gold label (label): based on the previous per-annotator labels.
- Probability of hallucination (p(Hallucination)): representing the proportion of annotators who deemed this specific datapoint to be a hallucination.
- Indicator of semantic reference (ref): specifying whether the target, source, or both contain the semantic information necessary to determine if a datapoint is a hallucination.

Furthermore, model-aware datapoints also identify the model used to produce each datapoint, represented by a Hugging Face identifier (model).

For each sample in the dataset, there is a source text, a target text and a hypothesis text. Depending on the task (DM, MT, PG) the goal is to determine if the Hypothesis contains any hallucination.

In the Table 1 there are three different samples from the official test set. Even if the labels are shown in the table along with the hallucination probabilities, during the evaluation phase of the competition, labels, and probabilities were hidden for the participants.

3 System Overview

Even if it has already been proved that the Transformers are not necessarily the best option for every text classification task (Siino et al., 2022), depending on the goal some strategies like domain-specific fine-tuning (Sun et al., 2019; Van Thin et al., 2023), or data augmentation (Lomonaco et al., 2023; Mangione et al., 2022; Siino et al., 2024a) can be beneficial for the considered task.

However, to address the task 6 hosted at SemEval-2024, we made use of a zero-shot learning approach (Chen et al., 2023; Wahidur et al., 2024), making use of the GPT Transformer named Llama 7B. This was dictated by our choice to bear in mind the computational efficiency without further feature engineering and/or heavy data preprocessing strategies.

Llama 2, a suite of large language models (LLMs), includes pretrained and fine-tuned models ranging from 7 to 70 billion parameters. Specifically tailored for dialogue applications, the fine-tuned LLMs are designated as Llama 2-Chat. The models demonstrate interesting performance when compared to open-source chat models across the majority of assessed benchmarks. Additionally, according to human evaluations focusing on helpfulness and safety, they could potentially serve as viable substitutes for closed-source models. Even if several others Open LLMs have proved to be able of outperforming Llama (Jiang et al., 2023), here we investigate the model's actual performance on this specific task. The authors of the model of-

[1] https://github.com/marco-siino/SemEval2024/

Target Text	Hypothesis Text	Label	p(Hallucination)
"Would you be surprised if I told you my name isn't actually Tom?"	*"You're gonna be surprised if I say my real name isn't Tom?"*	Not Hallucination	0.0
"There will be plenty of food."	*"The food will be full."*	Hallucination	0.8
"The two brothers are pretty different."	*"There's a lot of friends."*	Hallucination	1.0

Table 1: Three samples from the official test set are provided. Together with the labels for each sample, is also reported the probability of hallucination.

fer a comprehensive account of the fine-tuning approach and safety enhancements for Llama 2-Chat, with the aim of facilitating community engagement and contributing to the responsible advancement of LLM technology.

The Llama 2 suite comprises:

- Llama 2: an enhanced iteration of Llama 1, trained on a revised assortment of publicly available data. Notable improvements include a 40% augmentation in the size of the pretraining corpus, a doubling of the model's context length, and the adoption of grouped-query attention. Variants of Llama 2 with 7 billion, 13 billion, and 70 billion parameters are being released. Additionally, authors have trained 34 billion parameter variants, detailed in their paper but not released to the public;
- Llama 2-Chat: a fine-tuned version of Llama 2 tailored for dialogue applications.

To develop the new Llama 2 model family, the authors commenced with the pretraining methodology outlined in Touvron et al. 2023, utilizing an optimized autoregressive transformer. However, the authors made several modifications to enhance performance. These included more rigorous data cleaning, updates to data mixtures, training on 40% more total tokens, doubling the context length, and implementing grouped-query attention (GQA) to enhance inference scalability, particularly for larger models.

More specifically, given the task hosted at SemEval-2024, we asked the model: *"Is the Sentence supported by the Context above? Answer using ONLY yes or no:"*. To this request, the model replied with one or more words — usually starting with *yes* or *no* — that we parsed to extract one of the two labels. For example, given the context:

> *"The East African Islands are in the Indian Ocean off the eastern coast of Africa"*

The sentence:

> *"The eastern islands of the Indian Ocean are located in the eastern part of the Indian Ocean"*

And our question:

> *Is the Sentence supported by the Context above? Answer using ONLY yes or no:*

The model replied with:

> *no, the sentence is not supported by the context provided*

that we mapped into the label *Hallucination*.

It is worth noting that we needed to post-process the model answers to extract only the first word of the reply (i.e., *yes* or *no*). The model barely replied with a single word, even if prompted with the specific request of limiting its answer.

In the literature, several prompt engineering strategies have already been introduced (Denny et al., 2023; Giray, 2023). However, also from this perspective, we opted for a straight interaction with the GPT model, without any further engineering of the process. Finally, we collected all the predictions provided on the test set to into a JSON file with the required format to submit our predictions.

As noted in the recent study by Siino et al. 2024b, the contribution of preprocessing for text classification tasks is generally not impactful when using Transformers. More specifically, the best combination of preprocessing strategies is not very different from doing no preprocessing at all in the case of Transformers. For these reasons, and to keep our system highly fast and computationally light, we have not performed any preprocessing on the text.

4 Experimental Setup

We implemented our model on Google Colab. The library we used come from HuggingFace and as already mentioned is Llama 2[2]. Llama 2 comprises a series of pretrained and fine-tuned generative text models with parameter ranges spanning from 7 billion to 70 billion. This repository specifically hosts the 7B fine-tuned model, tailored for dialogue applications and converted to the Hugging Face Transformers[3] format. We also imported the Llama library (Touvron et al., 2023) from *llama_cpp*. The library is fully described on GitHub[4]. The dataset provided for all the phases are available on the Official Competition page. We did not perform any additional fine-tuning on the model. To run the experiment, a T4 GPU from Google has been used. After the generation of predictions, we exported the results on the format required by the organizers. As already mentioned, all of our code is available on GitHub.

5 Results

Submissions were divided into two tracks: a model-aware track, where organizers provide a checkpoint to a model publically available on Hugging Face for every data point considered, and a model-agnostic track, where organizers do not. The organizers encouraged participants to make use of model checkpoints in creative ways. For both tracks, all participants' submissions were evaluated using two criteria: the accuracy that the system reached on the binary classification; and the Spearman correlation of the systems' output probabilities with the proportion of the annotators marking the item as overgenerating. The evaluation script was made available[5], along with baseline systems and format checkers.

In the Table 2 we report the results obtained by our approach. In the rows are reported the two tracks (i.e., model agnostic or model aware) while in the column are reported the results according to the output score provided on CodaLab. As can be noted from the Tables 3, 4 our proposed approach it is not able to outperform the baseline provided for the task (i.e., Mistral 7B).

In the Table 3 and in the Table 4, the results obtained by the first three teams and by the last one, as showed on the official task page, are reported. Compared to the best performing models, our simple approach exhibits some room for improvements. Furthermore, our proposed approach is not able to outperform the baseline provided for the task. For this reason, we are confident that no further investigations should be performed for this task making use of the Llama model. However, it is worth notice that it required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab.

	Acc	Rho
Agnostic	0.625	0.204
Aware	0.671	0.244

Table 2: The method's performance on the test set. In the table are reported the results obtained by our private area on CodaLab.

6 Conclusion

This paper presents the application of a Llama-model for addressing the Task 6 at SemEval-2024. For our submission, we decided to follow a zero-shot learning approach, employing as-is, an in-domain pre-trained Transformer. After several experiments, we found beneficial to build a prompt containing the question for the model. Then we provide as a prompt the target sentence and the hypothesis sentence. The model was asked to decide if the hypothesis sentence is supported by the content of the target sentence, or if it is just a hallucinated text. The task is challenging, and there is still opportunity for improvement, as can be noted looking at the final ranking. Possible alternative approaches include utilizing the few-shot capabilities or also the use of other models like GPT and T5, increasing the size of the training set by using further data, or directly integrating other samples from the training and from the development sets. Further improvements could be obtained with a fine-tuning and modelling the problem as a text classification task. Furthermore, given the interesting results recently provided on a plethora of tasks, also other few-shot learning (Wang et al., 2023; Maia et al., 2024; Siino et al., 2023; Meng et al., 2024) or data augmentation strategies (Muftie and Haris, 2023; Tapia-Téllez and Escalante, 2020; Siino and Tinnirello, 2023) could be employed to improve the

[2] https://huggingface.co/TheBloke/Llama-2-7B-Chat-GGUF

[3] https://huggingface.co/

[4] https://github.com/ggerganov/llama.cpp

[5] https://helsinki-nlp.github.io/shroom/

TEAM NAME	ACC	RHO
GroupCheckGPT (1)	0.847	0.769
OPDAI (2)	0.836	0.732
HIT_WL (3)	0.831	0.768
baseline system	0.697	0.403
OxYuan (48)	0.461	0.134

Table 3: Comparing performance on the test set for the model agnostic track. In the table are shown the results obtained by the first three teams and by the last one. In parentheses is reported the position in the official final ranking.

TEAM NAME	ACC	RHO
HaRMoNEE (1)	0.813	0.699
GroupCheckGPT (2)	0.806	0.715
TU Wien (3)	0.806	0.707
baseline system	0.745	0.488
octavianB (45)	0.483	-0.064

Table 4: Comparing performance on the test set for the model aware track. In the table are shown the results obtained by the first three users and by the last one. In parentheses is reported the position in the official final ranking.

results. Looking at the final ranking, our simple approach exhibits some room for improvements. However, it is worth notice that required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab.

Acknowledgments

We extend our gratitude to the anonymous reviewers for their insightful comments and valuable suggestions, which have significantly enhanced the clarity and presentation of this paper.

References

Shiming Chen, Ziming Hong, Wenjin Hou, Guo-Sen Xie, Yibing Song, Jian Zhao, Xinge You, Shuicheng Yan, and Ling Shao. 2023. Transzero++: Cross attribute-guided transformer for zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12844 – 12861.

Paul Denny, Viraj Kumar, and Nasser Giacaman. 2023. Conversing with copilot: Exploring prompt engineering for solving cs1 problems using natural language. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 1136–1142.

Louie Giray. 2023. Prompt engineering with chatgpt: a guide for academic writers. *Annals of biomedical engineering*, 51(12):2629–2633.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Jiarui Li, Ye Yuan, and Zehua Zhang. 2024. Enhancing llm factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. *arXiv preprint arXiv:2403.10446*.

Francesco Lomonaco, Marco Siino, and Maurizio Tesconi. 2023. Text enrichment with japanese language to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2708–2716. CEUR-WS.org.

Beatriz Matias Santana Maia, Maria Clara Falcão Ribeiro de Assis, Leandro Muniz de Lima, Matheus Becali Rocha, Humberto Giuri Calente, Maria Luiza Armini Correa, Danielle Resende Camisasca, and Renato Antonio Krohling. 2024. Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer. *Expert Systems with Applications*, 241:122418.

Stefano Mangione, Marco Siino, and Giovanni Garbo. 2022. Improving irony and stereotype spreaders detection using data augmentation and convolutional neural network. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2585–2593. CEUR-WS.org.

Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (llm) hallucination. In *European Semantic Web Conference*, pages 182–185. Springer.

Zong Meng, Zhaohui Zhang, Yang Guan, Jimeng Li, Lixiao Cao, Meng Zhu, Jingjing Fan, and Fengjie Fan. 2024. A hierarchical transformer-based adaptive metric and joint-learning network for few-shot rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(3).

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Fuad Muftie and Muhammad Haris. 2023. Indobert based data augmentation for indonesian text classification. In *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, page 128 – 132.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2022. Fake news spreaders detection: Sometimes attention is not all you need. *Information*, 13(9):426.

Marco Siino, Francesco Lomonaco, and Paolo Rosso. 2024a. Backtranslate what you are saying and i will tell who you are. *Expert Systems*, n/a(n/a):e13568.

Marco Siino, Maurizio Tesconi, and Ilenia Tinnirello. 2023. Profiling cryptocurrency influencers with few-shot learning using data augmentation and ELECTRA. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2772–2781. CEUR-WS.org.

Marco Siino and Ilenia Tinnirello. 2023. Xlnet with data augmentation to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2763–2771. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024b. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

José Medardo Tapia-Téllez and Hugo Jair Escalante. 2020. Data augmentation with transformers for text classification. In *Advances in Computational Intelligence*, pages 247–259, Cham. Springer International Publishing.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Dang Van Thin, Duong Ngoc Hao, and Ngan Luu-Thuy Nguyen. 2023. Vietnamese sentiment analysis: An overview and comparative study of fine-tuning pretrained language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–27.

Rahman S. M. Wahidur, Ishmam Tashdeed, Manjit Kaur, and Heung-No Lee. 2024. Enhancing zero-shot crypto sentiment with fine-tuned language model and prompt engineering. *IEEE Access*, 12:10146 – 10159.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Trans. Circuits Syst. Video Technol.*, 33(12):7789–7802.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

DKE-Research at SemEval-2024 Task 2: Incorporating Data Augmentation with Generative Models and Biomedical Knowledge to Enhance Inference Robustness

Yuqi Wang[1,3], Zeqiang Wang[4], Wei Wang[1], Qi Chen[1], Kaizhu Huang[2], Anh Nguyen[3], Suparna De[4]

[1]Xi'an Jiaotong Liverpool University [2]Duke Kunshan University [3]University of Liverpool [4]University of Surrey

yuqi.wang17@student.xjtlu.edu.cn, {wei.wang03,qi.chen02}@xjtlu.edu.cn,

kaizhu.huang@dukekunshan.edu.cn, anh.nguyen@liverpool.ac.uk,

{zeqiang.wang, s.de}@surrey.ac.uk

Abstract

Safe and reliable natural language inference is critical for extracting insights from clinical trial reports but poses challenges due to biases in large pre-trained language models. This paper presents a novel data augmentation technique to improve model robustness for biomedical natural language inference in clinical trials. By generating synthetic examples through semantic perturbations and domain-specific vocabulary replacement and adding a new task for numerical and quantitative reasoning, we introduce greater diversity and reduce shortcut learning. Our approach, combined with multi-task learning and the DeBERTa architecture, achieved significant performance gains on the NLI4CT 2024 benchmark compared to the original language models. Ablation studies validate the contribution of each augmentation method in improving robustness. Our best-performing model ranked 12th in terms of faithfulness and 8th in terms of consistency, respectively, out of the 32 participants.

1 Introduction

In the domain of clinical trial analysis, researchers and practitioners are overwhelmed with an ever-expanding corpus of clinical trial reports (CTRs). The current repository contains a vast number of documents and is rapidly growing, a trend that correlates with the increasing prevalence of cross-national, cross-ethnic, and multi-center clinical studies (Bastian et al., 2010). This growth necessitates a scalable approach to evaluate and interpret the massive amount of data in these reports (Goldberg et al., 2017; Li and Bergan, 2020).

Recent advances in Natural Language Processing (NLP) offer promising avenues for the automated analysis of CTRs. Such analyses include medical evidence understanding (Nye et al., 2021), information retrieval (Wang et al., 2023b), causal relationship identification (Cai et al., 2017), and the inference of underlying reasons for trial outcomes (Steinberg et al., 2023). Integrating natural language inference (NLI) with CTRs has the potential to revolutionize the large-scale, NLP-based examination of experimental medicine (Kim and Delen, 2018). Despite the progress in NLP, the application of large language models to this task presents several challenges, including susceptibility to shortcut learning, hallucination, and biases stemming from word distribution patterns within the training data (Huang et al., 2023).

To address these issues, we propose a novel method that leverages generative language models, such as GPT-3.5[1], and biomedical domain knowledge graphs to enhance data diversity. Our approach introduces three types of data augmentation: numeric question-answering data generation, semantic perturbations, and domain-tailored lexical substitutions for the biomedical field. By combining these data augmentation techniques with multi-task learning and the DeBERTa (He et al., 2021) architecture, we have achieved significant improvements in terms of faithfulness and consistency on the NLI4CT 2024 dataset. This paper outlines our approach, elaborates on the design of the perturbations and the multi-task learning process, and demonstrates the efficacy of our method through rigorous evaluation.

2 Background

In a crucial field like healthcare, where misinterpretations can have severe implications, NLI models must present precise predictions and reliable interpretations. This highlights the importance of accurate and trustworthy reasoning in these NLI models.

SemEval 2024 Task 2 (Jullien et al., 2024) provides multi-sentence textual data consisting of patient case histories and medical reports. The objec-

[1]https://openai.com/chatgpt

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 88–94

tive of this task is to predict the logical relationship between the CTR and a given statement, including entailment and contradiction. The evaluation emphasizes prediction accuracy as well as the robustness to the controlled interventions, helping increase healthcare practitioners' trust in the system's predictions.

Enhancing the robustness of NLI models for healthcare can be strategically achieved using data augmentation techniques. Synthetic data generation via techniques like conditional text generation can expand training data diversity and volume to improve model generalization capabilities (Liu et al., 2020; Puri et al., 2020; Bayer et al., 2023). Meanwhile, multi-task learning with auxiliary objectives related to logical reasoning and explanation generation can enhance faithful reasoning abilities (Li et al., 2022). Useful domain knowledge can be captured by training language models on domain-specific medical textual datasets (Singhal et al., 2023; Tian et al., 2024). Complementary data-centric methods can augment model architecture design to develop more capable, trustworthy, and clinical NLI systems.

3 System overview

In this section, we describe the proposed system to tackle the NLI problem and enhance the model's robustness against interventions spanning numerical, vocabulary, and semantic dimensions, as shown in Figure 1.

3.1 Data for Numerical Question Answering Task

A major limitation of many language models lies in their tendency to learn linguistic patterns and features from large-scale textual data while lacking capabilities for numerical and quantitative reasoning (Geva et al., 2020). Such capabilities are crucial for analyzing relationships between CTRs and corresponding claims. Although BERT-based models pre-trained on NLI tasks, i.e. DeBERTa, can conduct general linguistic inference, they remain vulnerable to numerical perturbations in statements.

Therefore, we propose to leverage GPT-3.5 to generate data tailored to the numerical question-answering task based on original entailed statements: The entailed statement, denoted as x, corresponding to a given CTR, is converted into a question q that requires numerical reasoning. Subsequently, three candidate choices c are enumerated, each accompanied by an answer a extracted from the original statements. The loss function employed for this task is binary cross-entropy and is expressed as follows:

$$\mathcal{L}_{NQA} = \begin{cases} -\log g_\theta\left(\text{CTR}, q, c;\right) & c = a \\ -\left[1 - \log g_\theta\left(\text{CTR}, q, c\right)\right] & c \neq a \end{cases}$$

where $g(\cdot)$ is the function to determine if the candidate choice is the correct answer, and θ is the corresponding parameters for the DeBERTa backbone network and the additional classifier.

This numerical question-answering task serves as an auxiliary task to enhance numerical reasoning abilities. The final loss function for the system combines the losses from this task and the main NLI task, i.e.

$$\mathcal{L} = \mathcal{L}_{NLI} + \lambda \mathcal{L}_{NQA}$$

where λ is the hyper-parameter to be tuned in the validation phase.

3.2 Semantic Perturbation

We utilize GPT-3.5 to generate perturbed statements based on the original entailed input, obtaining both semantic-altering variants labeled as "contradictions" and semantic-preserving variants labeled as "entailment". Specifically, to produce contradictory versions, guiding keywords such as "contradicted" and "minor changes" are injected into the input prompt to slightly modify the original statement while altering the semantics to create a contradiction. Conversely, to generate entailed versions, guiding phrases such as "paraphrase" are included in the prompt to rephrase the statement extensively while retaining semantic equivalence. This controlled semantic perturbation of the input statement via guided text generation allows us to efficiently augment the dataset with both contradicting and entailing variants of the original input.

3.3 Vocabulary Replacement

When we analyze textual data in the clinical domain, we need to pay attention to the vocabulary because it contains many terms that are specific to this domain (Wang et al., 2018). However, most NLI models are pre-trained on data from general domains, and they are unaware of the meaning or relevance of these terms (Wang et al., 2023a). To address this problem, we use a combination of biomedical knowledge graph embedding and statistical model, which can help us find the most important keyword to replace the term in the statement

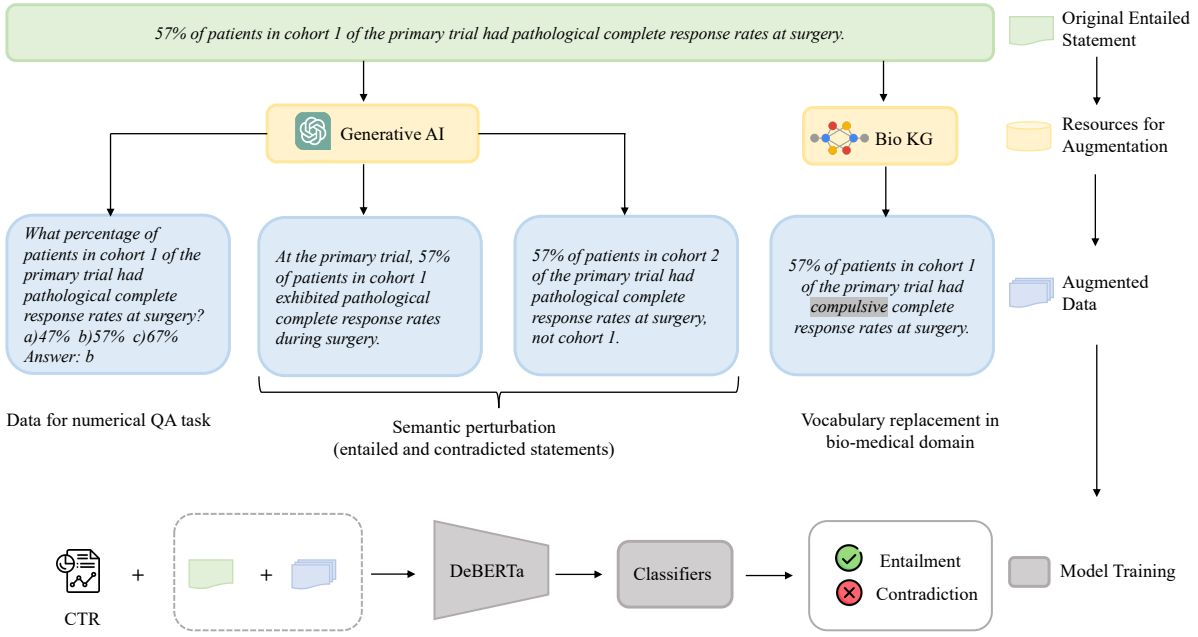


Figure 1: The overall demonstration of the proposed system. The upper part of the demonstration involves the application of data augmentation techniques to entailed statements extracted from the original NLI dataset, leveraging generative artificial intelligence (AI) and biomedical domain knowledge graphs. Specifically, we undertake the following procedures: 1) Transformation of statements into multiple-choice questions accompanied by corresponding answers; 2) Introduction of semantic perturbations to the original entailed statements; 3) Employing a statistical method to identify keywords within the original entailed statements, followed by their substitution with synonyms sourced from the biomedical knowledge graph. In the lower part of the demonstration, we incorporate the original entailed statements, augmented data, and CTRs as training data to develop a classifier based on the DeBERTa architecture.

and generate the augmented data to improve the vocabulary alignment. Specifically, given a statement x, consisting of n words, i.e. $x = \{w_1, w_2, ..., w_n\}$ and the set of all the statements, denoted as D, we first remove all the stop-words and apply Term-Frequency-Inverse Document Frequency (TF-IDF) to identify the most important term in the statement, i.e.

$$w^* = \arg\max_{w_i \in x} \text{TF}(w_i, x) \times \text{IDF}(w_i, D)$$

Subsequently, we locate a term in the biomedical embedding space that shares the same part-of-speech and has the highest similarity score with the chosen term, using it as the substitute, i.e.

$$\hat{w}^* = \arg\max_{w \in V} \{sim(w^*, w) \mid \text{PoS}(w) = \text{PoS}(w^*)\}$$

where V is the biomedical term vocabulary and $\text{PoS}(\cdot)$ is the part-of-speech of a word. In this way, we can substitute w^* in the original statement with $\hat{w}^*$ to generate a new adversarial sample to enhance the model robustness in the vocabulary aspect.

4 Experimental setup

4.1 Dataset

	Ent.	Con.	Alt.	Pres.	SUM
Train	850	850	-	-	1,700
Val.	100	100	1,606	336	2,142
Test	250	250	4,136	864	5,500

Table 1: Statistics of the validation and test set. "Ent." and "Con." stands for entailment and contradiction, while "Alt." and "Pres." stands for altering and preserving.

We conducted experiments on the NLI4CT 2024 dataset (Jullien et al., 2024), generated by clinical domain experts and sourced from a large database for clinical studies[2]. The statistic of this dataset is summarized in Table 1. The training data is the same as the NLI4CT 2023 dataset (Jullien et al., 2023) while there are perturbed samples in the validation and testing sets.

[2]https://ClinicalTrials.gov

4.2 Metrics

We first assessed the performance of the original statements without any perturbation and recorded the corresponding F1 score, precision, and recall. Then, we assessed the performance of the contrast set, consisting of interventions. Specifically, to evaluate the model's robustness to the semantic-preserving interventions, we used consistency as the metric, i.e.

$$Consistency = \frac{1}{N}\sum_{1}^{N} 1 - \left|f(x_i') - f(x_i)\right|$$
$$x_i' \in C : \text{Label}(x_i) = \text{Label}(x_i')$$

Where C is the contrast set, and N is the number of the statements in the contrast set. x_i' is the perturbed statement for x_i and $f(\cdot)$ computes the final prediction from the model. For the semantic-altering interventions, we evaluated the model using faithfulness, i.e.

$$Faithfulness = \frac{1}{N}\sum_{1}^{N} \left|f(x_i') - f(x_i)\right|$$
$$x_i' \in C : \text{Label}(x_i) \neq \text{Label}(x_i'), \text{ and } f(x_i) = \text{Label}(x_i)$$

4.3 Implementation details

We downloaded DeBERTa models from the Huggingface repository[3] and implemented our proposed method based on Python 3.10 and Pytorch 2.1.1. During the model training, we used the Adam optimizer and set the learning rate to $5e-6$ with a batch size of 4, following the original work (He et al., 2021). The maximum sequence length the model can take was set to 512. The epoch number was set to 20, and the early stopping based on the validation set was applied to avoid overfitting. The input format for the NLI task in this work is structured as follows: [CLS] + CTR + [SEP] + claim + [SEP]. In this structure, [CLS] serves as the initial token for classification in DeBERTa, and [SEP] acts as a separator token. For the vocabulary replacement, we used the bio-medical domain embedding from the work by (Zhang et al., 2019), which has been pre-trained over the MeSH knowledge graph[4]. For preprocessing, such as stop word filtering and part-of-speech tagging, we used the NLTK library[5] in Python. We include prompts for numerical question-answering data generation and semantic perturbation in Table 2.

[3]https://huggingface.co/

[4]https://www.ncbi.nlm.nih.gov/mesh/

[5]https://www.nltk.org/

5 Results

We conducted experiments with different-sized DeBERTa models, iteratively adding augmented data from three different interventions to the training set. As shown in Table 3, incorporating all three types of augmented data greatly improved the average faithfulness and consistency scores. Specifically, we witnessed gains of 8.17% on DeBERTa-l and 2.37% on DeBERTa-b. This result also suggests that the augmented training data provided more benefit to the larger-sized DeBERTa model in terms of robustness. The additional augmented examples may have provided useful regularization, helping it generalize better on both the unaltered control and contrast datasets. Our best-performing model ranked 12th in terms of faithfulness and 8th in terms of consistency, respectively, out of the 32 participants.

From this iterative process, we can see that semantic perturbation with generative AI contributes mainly to the performance gain for both NLI models. Compared with this, vocabulary replacement in the biomedical domain has only a minor effect. This may suggest that vocabulary replacement in our work may be relatively less effective in this case because it only swaps out individual words, while semantic perturbation modifies the whole statement. Hence, semantic perturbation provides more meaningful variations to augment the training data.

While the augmented data improved the robustness to interventions, we noticed a slight performance drop in the control set. For example, the F1 score on the control set decreased by 3.16% for DeBERTa-l and 0.48% for DeBERTa-b after adding all the augmented data. This performance decline indicates there may have been a small trade-off between improving robustness to interventions and maintaining strong performance on the original data. One of the reasons accounting for this could be that the generative AI may generate noisy or irrelevant data. For example, in numerical question answering data generation, if the original entailed statement discusses an assumption about a 50-year-old patient not mentioned in the CTR, the generative model may create an unrelated question about the patient's age that cannot be inferred from the given information. Another example involves vocabulary replacement: we observed that there exist some cases where even two words having very similar embeddings in the biomedical domain

	Prompt
NQA	*Please convert the statement to a multiple choice question that requires the numerical or quantitative reasoning, and each question has 3 choices, using the given template: \n Question: [Question] \n Choices: 1. [Choice 1]\n 2. [Choice 2]\n 3. [Choice 3]\n Correct Answer: [Correct Answer].*
SP.-Ent.	*Please rephrase the given statement:*
SP.-Con.	*Please generate a contradictory statement based on the given statement, with a minor change:*

Table 2: Prompts for numerical question-answering data generation and semantic perturbation. NQA stands for numerical question answering. SP.-Con. and SP.-Ent. means semantic perturbation to generate statements labeled as contradiction and entailment, respectively.

Method	Validation					Test				
	F1	Prec.	Rec.	Faith.	Con.	F1	Prec.	Rec.	Faith.	Con.
DeBERTa-l	**81.82**	90.00	75.00	73.81	71.48	**77.25**	80.80	73.99	67.13	71.06
+SP	81.77	83.00	80.58	85.42	**75.16**	75.52	72.80	78.45	78.24	74.01
+VR	81.00	81.00	81.00	86.01	74.16	75.05	71.60	78.85	78.59	74.42
+NQA	80.60	81.00	80.20	**86.61**	74.91	74.09	69.20	79.72	**79.98**	**74.54**
DeBERTa-b	70.87	73.00	68.87	49.40	60.02	**62.53**	60.40	64.81	57.75	59.33
+SP	**71.84**	74.00	69.81	51.49	60.65	62.08	59.60	64.78	60.65	59.70
+VR	70.59	72.00	69.23	52.38	60.71	62.21	59.60	65.07	60.76	59.72
+NQA	70.30	71.00	69.61	**52.98**	**60.77**	62.05	59.20	65.20	**61.92**	**59.89**

Table 3: Results on the development set and testing set for NLI4CT 2024 dataset. DeBERTa-l and DeBERTa-b are the large version and base version of the DeBERTa model, respectively. SP and VR stand for semantic perturbation and vocabulary replacement. The best results for F1 score on the control set, faithfulness, and consistency are highlighted.

knowledge graph embedding space may not be very closely related in the context of the current statement. Including these illogical examples in the augmented training data could mislead the original DeBERTa model, resulting in worse performance on the unaltered control set.

6 Conclusion

In this work, we proposed a data augmentation approach to enhance the robustness of natural language inference models for clinical trial report analysis. Our method leverages generative AI and biomedical knowledge graphs to augment training data along three dimensions: numerical reasoning, semantic perturbations, and domain-tailored lexical substitutions. Experiments on the NLI4CT 2024 dataset demonstrate that our approach effectively improves model faithfulness and consistency against controlled interventions, with significant gains against the DeBERTa baselines.

However, we observed a slight performance drop on the unaltered test set, indicating a trade-off between robustness to perturbations and maintaining strong performance on original data. Future work will focus on: 1) generating higher-quality augmented examples using numerical question-answering data generation to minimize or avoid performance drop; 2) validating the perturbed samples to help remove noisy or irrelevant examples (Wang et al., 2023c); 3) incorporating external structured knowledge via pre-training on knowledge graphs and not just lexical substitution, which can provide more contextual domain information.

7 Acknowledgments

We would like to thank all the anonymous reviewers for their valuable feedback. We would like to acknowledge the financial support provided by the

Postgraduate Research Scholarship (PGRS) (contract number PGRS-20-06-013) at Xi'an Jiaotong-Liverpool University. Additionally, this research has received partial funding from the Jiangsu Science and Technology Programme (contract number BK20221260) and the Research Development Fund (contract number RDF-22-01-132) at Xi'an Jiaotong-Liverpool University.

References

Hilda Bastian, Paul Glasziou, and Iain Chalmers. 2010. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9):e1000326.

Markus Bayer, Marc-André Kaufhold, Björn Buchhold, Marcel Keller, Jörg Dallmeyer, and Christian Reuter. 2023. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International journal of machine learning and cybernetics*, 14(1):135–150.

Ruichu Cai, Mei Liu, Yong Hu, Brittany L Melton, Michael E Matheny, Hua Xu, Lian Duan, and Lemuel R Waitman. 2017. Identification of adverse drug-drug interactions through causal association rule discovery from spontaneous adverse event reports. *Artificial intelligence in medicine*, 76:7–15.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958.

Richard M Goldberg, Lai Wei, and Soledad Fernandez. 2017. The evolution of clinical trials in oncology: defining who benefits from new drugs using innovative study designs.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Yong-Mi Kim and Dursun Delen. 2018. Medical informatics research trend analysis: A text mining approach. *Health informatics journal*, 24(4):432–452.

Allen Li and Raymond C Bergan. 2020. Clinical trial design: Past, present, and future in the context of big data and precision medicine. *Cancer*, 126(22):4838–4846.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu. 2022. Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods. *arXiv preprint arXiv:2203.05227*.

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020. Data boost: Text data augmentation through reinforcement learning guided conditional generation. *arXiv preprint arXiv:2012.02952*.

Benjamin E Nye, Jay DeYoung, Eric Lehman, Ani Nenkova, Iain J Marshall, and Byron C Wallace. 2021. Understanding clinical trial reports: Extracting medical entities and their relations. *AMIA Summits on Translational Science Proceedings*, 2021:485.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Ethan Steinberg, Nikolaos Ignatiadis, Steve Yadlowsky, Yizhe Xu, and Nigam Shah. 2023. Using public clinical trial reports to probe non-experimental causal inference methods. *BMC Medical Research Methodology*, 23(1):204.

Shubo Tian, Qiao Jin, Lana Yeganova, Po-Ting Lai, Qingqing Zhu, Xiuying Chen, Yifan Yang, Qingyu Chen, Won Kim, Donald C Comeau, et al. 2024. Opportunities and challenges for chatgpt and large language models in biomedicine and health. *Briefings in Bioinformatics*, 25(1):bbad493.

Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. 2018. A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*, 87:12–20.

Yuqi Wang, Wei Wang, Qi Chen, Kaizhu Huang, Anh Nguyen, Suparna De, and Amir Hussain. 2023a. Fusing external knowledge resources for natural language understanding techniques: A survey. *Information Fusion*, 92:190–204.

Yuqi Wang, Zeqiang Wang, Wei Wang, Qi Chen, Kaizhu Huang, Anh Nguyen, and Suparna De. 2023b. Zero-shot medical information retrieval via knowledge graph embedding. In *International Workshop on Internet of Things of Big Data for Healthcare*, pages 29–40. Springer.

Zimu Wang, Wei Wang, Qi Chen, Qiufeng Wang, and Anh Nguyen. 2023c. Generating valid and natural adversarial examples with large language models. *arXiv preprint arXiv:2311.11861*.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019. Biowordvec, improving biomedical word embeddings with subword information and mesh. *Scientific data*, 6(1):52.

SATLab at SemEval-2024 Task 1: A Fully Instance-Specific Approach for Semantic Textual Relatedness Prediction

Yves Bestgen
Statistical Analysis of Text Laboratory (SATLab)
Université catholique de Louvain
Place Cardinal Mercier, 10 1348 Louvain-la-Neuve, Belgium
yves.bestgen@uclouvain.be

Abstract

This paper presents the SATLab participation in SemEval 2024 Task 1 on Semantic Textual Relatedness. The proposed system predicts semantic relatedness by means of the Euclidean distance between the character ngram frequencies in the two sentences to evaluate. It employs no external resources, nor information from other instances present in the material. The system performs well, coming first in five of the twelve languages. However, there is little difference between the best systems.

1 Introduction

Semantic similarity between words, phrases and texts has long attracted the attention of NLP researchers. It is obviously a useful source of information in tasks such as information retrieval, text summarization, question answering or machine translation (Agirre et al., 2012). It has been the subject of several shared tasks within SemEval since 2012 (Agirre et al., 2012; Marelli et al., 2014; Cer et al., 2017). More recently, interest has also focused on Semantic Textual Relatedness (STR), which is supposed to be a more general concept. As Abdalla et al. (2023) point out, two sentences must be paraphrases or present an entailment relation to be semantically similar, whereas to be related, it is sufficient that they deal with similar themes or express similar points of view on a given issue. Work on STR is less advanced due to the lack of annotated datasets on this dimension (Abdalla et al., 2023). It should be noted, however, that the human annotators who evaluated semantic textual similarity for the SILK dataset (Marelli et al., 2014) clearly evaluated relatedness, since they considered pairs of sentences that contradict each other as semantically very similar (96% similarity), such as in SILK Instance 466:

- *A man is performing a trick on a green bicycle.*
- *There is no man performing a trick on a green bicycle.*

The SILK dataset contains many other examples of this kind of judgement. This observation suggests that the term "relatedness" is more appropriate to describe this field of research, at least when dealing with the intuition of native speakers. It also suggests that techniques that are effective in automatically estimating semantic similarity should also be effective in estimating relatedness. These are mainly state-of-the-art deep learning algorithms (Cer et al., 2017).

In this context, Ousidhoum et al. (2024b) have proposed the SemEval 2024 Task 1, which has a number of specific features compared with previous work. Firstly, the task focuses on relatedness, and is based on material consisting of sentence pairs that have been annotated on this dimension by native speakers. Secondly, the task is highly multilingual, covering more than ten languages, some of which are very poorly resourced. Finally, it includes three subtasks: supervised, unsupervised and crosslingual. In the supervised subtask, the systems were to be trained using training datasets provided by the task organizers. In the unsupervised subtask, no datasets labeled according to semantic relatedness or semantic similarity could be used. In the crosslingual subtask, the system had to be trained on a language other than the target language.

2 The Proposed Approach

Due to its highly multilingual nature (twelve languages), the unsupervised subtask seemed a priori to be particularly interesting for the development of a generic approach, as language-independent as possible. This would be the case of a system that estimates the semantic relatedness of a pair of sentences without recourse to any resources external to the material and even without taking into account the other instances present in the material. A system takes other instances into account when, for example, it weights an instance features according

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 95–100

to their frequency in the complete material, using the classic TF-IDF. A system is completely independent of other instances when the processing of one instance is not affected in any way by the other instances it has to predict. The system proposed by the SATLab fulfills this requirement by using the Euclidean distance between the two sentences, calculated on the basis of the frequency of the ngrams of characters that make them up. If such a system proves successful to predict semantic relatedness, it could become a potential candidate for the analysis of any language.

Admittedly, such a system is more akin to a baseline than a state-of-the-art system. However, it should also be noted that systems based on character ngrams have for many years been considered particularly effective for NLP tasks such as language identification, error correction, information retrieval and even for hate speech and offensive content identification (Damashek, 1995; Bestgen, 2021b). Character ngrams have the advantage of not requiring material to be tokenized, which can be problematic in some Asian languages, and of being able to extract morphological information at very low cost (Peng et al., 2003).

This paper presents SATLab's participation in SemEval 2024 Task 1 with this fully instance-specific system. The following section introduces the task and describes the proposed system. The results obtained are then reported.

3 The Unsupervised Task

Subtask 1B of SemEval 2023 (Ousidhoum et al., 2024b) asked participating teams to estimate the semantic relatedness between pairs of sentences in twelve languages: five Afro-Asiatic (Algerian Arabic [arq], Amharic [amh], Hausa [hau], Modern Standard Arabic [arb], Moroccan Arabic [ary]), five Indo-European (Afrikaans [afr], English [eng], Hindi [hin], Punjabi [pan] and Spanish [spa]), one Austronesian (Indonesian [ind]) and one from the Niger-Congo family (Kinyarwanda [kin]). The material, collected by Ousidhoum et al. (2024a), was selected from various resources such as semantic similarity datasets, news articles and Wikipedia texts. After this material had been carefully checked, it was submitted to native speakers whose task was to assess the semantic relatedness between pairs of sentences using the Best-Worst Scaling procedure. Ousidhoum et al. (2024a) reported high to near-perfect inter-rater reliabilities (split-half correlations: Min = 0.64, Max = 0.96).

In this Task 1B, the systems had to be unsupervised, since no dataset including evaluations of semantic relatedness between sentence pairs or texts could be employed. It should be noted, however, that the organizers provided participants with development data similar to that provided later for the testing phase, and that a team's predictions for these data could be evaluated by submitting them to CodaLab. The few tests I carried out showed that performance varied greatly depending on the language. It therefore didn't seem advisable to rely on this development material to make general decisions about the system to be developed. In the testing phase, only one prediction for each language could be submitted, and the performance measure was Spearman's rank correlation coefficient.

4 The SATLab System

A single system was used for all twelve languages. It is adapted from the one developed for the authorship identification of source code (Bestgen, 2020). This system takes as input each pair of utterances and outputs a distance between them without any other information, either from the rest of the material or external to it. Each pair of utterances is therefore processed in a way that is completely independent of the other pairs present in the material.

The only pre-processing is the lower-casing of all texts as included in SAS. I have to admit that it's not obvious to me what impact this has on languages as unknown to me as Kinyarwanda or Amharic. No tokenization or lemmatization has been applied. The system uses character ngrams made up of 1 to 5 characters. All characters are taken into account, including spaces, punctuation marks, symbols, characters from other writing systems, etc. The ngrams at the beginning and end of each statement are distinguished from the others. All ngrams in a statement are retained, so there is no frequency threshold. The frequency of each feature is weighted by a logarithmic function using the formula: $1 + \log(Freq)$. Finally, the features of each statement are weighted by the L2 norm (thus instance-wise). Most of these system components have be taken from the one developed for a difficult language identification problem (Bestgen, 2021a).

The Euclidean distance between the sets of ngrams of each utterance in a pair is used to estimate the semantic dissimilarity between these utterances. Before submission, these distances are

transformed into similarity by ranking them from largest to smallest. No information is lost through such ranking, since the organizers have chosen a rank correlation as the efficiency criterion.

5 Analysis and Results

5.1 Official Results

Twelve teams took part in the test phase of Task 1B, but only five proposed solutions for all twelve languages. One team proposed a solution for all languages except Spanish. The organizers provided a baseline based on the number of shared words between the two sentences of a pair (SemRel Lexical Overlap Baseline, see Ousidhoum et al. (2024b) for details).

Figure 1 shows the performance of all the systems for the twelve languages, highlighting the baseline and the system proposed by the SATLab. Marks not connected by a line are from systems that did not submit a solution for all languages. I don't know whether the systems proposed by the other teams are identical for all twelve languages, as is the case for the baseline and the SATLab.

This figure merits several comments. Firstly, when we analyze the overall results, we observe that the profiles of the teams[1] who submitted for all languages are similar. This observation is confirmed by an analysis of the Pearson correlations between these profiles. The lowest correlation is 0.54, only two are below 0.63 and half of them are above 0.73. These profiles highlight strong variations in performance according to language. While almost all the teams performed well to very well for Afrikaans (afr), Amharic (amh), English (eng) and Spanish (spa), they performed poorly for Punjabi (pan), with the SATLab system even achieving a negative correlation. It therefore appears that the material for some languages is considerably more complicated than for others. A detailed analysis of the differences between these materials would therefore be very useful.

Figure 1 also shows that the SATLab's performance is as good as or better than that of other teams in the vast majority of languages, but there is little difference between the best teams. This second observation would certainly be confirmed if confidence intervals, obtained by bootstrapping (Bestgen, 2022), were presented, but their calculation requires access to the predictions of all systems. In any case, when performances are so close, it is essential to take into account other factors such as computational complexity, which will be possible when reading the system descriptions of the other teams.

Finally, Figure 1 also shows that the organizers' baseline is superior to all other systems for two languages: Hindi (hin) and Moroccan Arabic [ary]. Clearly, this is an underperformance by all participants.

5.2 System Component Analysis

To assess the contribution of each component to the system overall performance, all of them were modified, one at a time, and the system was re-evaluated using the gold standard provided by the task organizers for eleven languages. The results are shown in Table 1 using the difference between each modified system and the official SATLab system, whose performance is shown in the first row.

The only pre-processing of the material carried out, the lower casing, brings benefits in only two languages. Presumably, it doesn't affect the many languages that don't use Latin characters. Using ngrams whose maximum length is one character shorter or one character longer has very little impact. On the other hand, feature weighting by TF-IDF is beneficial in ten out of eleven languages. Not using L2 normalization profoundly alters performance. While it brings significant benefit in one language, the impact is negative in nine languages, and can reach -0.574. As far as distance is concerned, Dice is more efficient than the Euclidean distance, but the gain is significantly lower than that obtained by applying the Euclidean distance to the weights transformed by TF-IDF.

The last line gives the correlations obtained by the system when TF-IDF is used instead of the logarithmic weighting. The gains over the official SATLab submission are sufficiently large to conclude that a fully instance-specific approach is significantly less effective at predicting STR than an approach that takes into account the other instances of the test material (which TF-IDF does, as explained in the introduction). There is no point in comparing these correlations with those of the other participants, since they would certainly have submitted a different system if they had been able to optimize it as just done.

[1] In this discussion of results, the baseline is considered a "team".

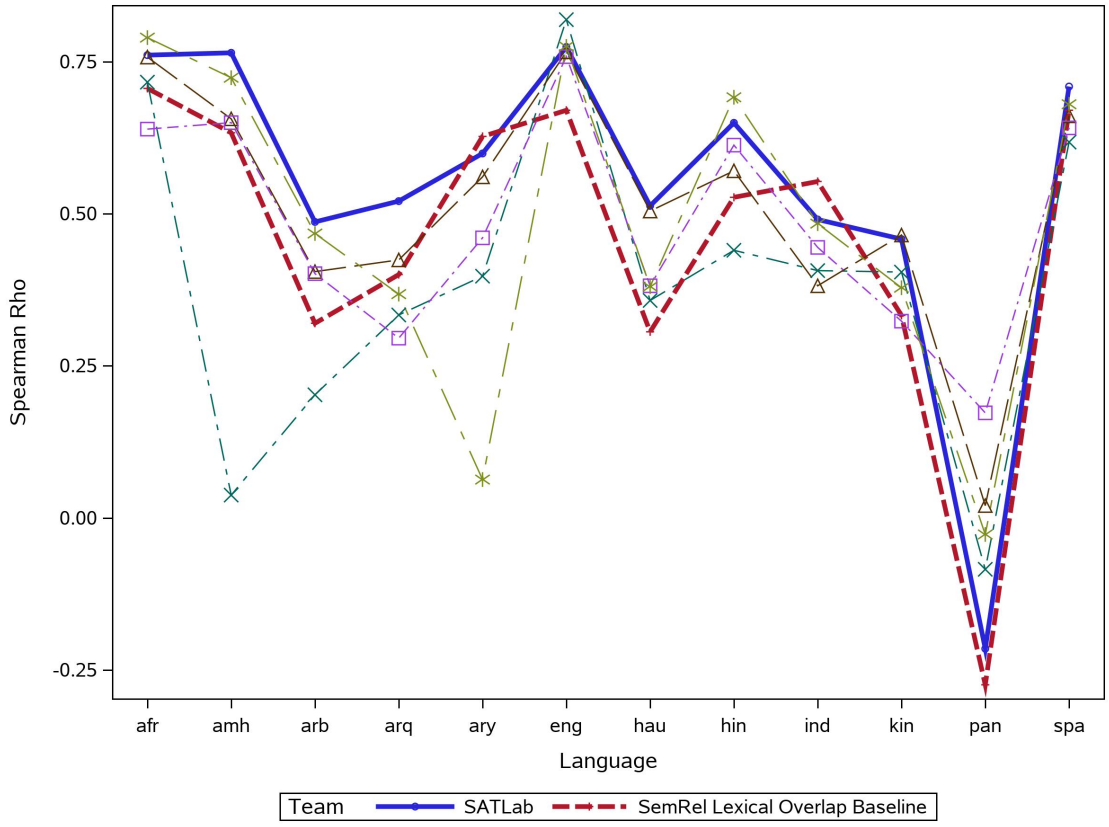


Figure 1: Performances of all systems for the twelve languages

Expe	afr	amh	arb	arq	ary	eng	hau	hin	ind	kin	pan
Submitted	0.761	0.764	0.487	0.521	0.599	0.774	0.513	0.649	0.491	0.458	-0.215
No Lowercase	0.005	0.000	0.000	0.000	0.000	-0.02	-0.028	0.000	0.005	0.005	0.000
4-grams	-0.003	-0.001	-0.016	0.005	-0.012	-0.002	-0.015	-0.007	0.007	0.012	0.018
6-grams	0.001	0.000	0.004	-0.009	-0.002	-0.001	0.003	0.000	-0.012	-0.004	-0.009
TF-IDF	0.021	0.001	0.061	0.052	0.024	0.024	0.057	0.046	-0.052	0.069	0.002
BM25	0.011	-0.008	0.043	-0.085	0.005	0.014	0.026	-0.091	-0.077	0.083	0.032
No L2	-0.144	-0.247	-0.421	-0.574	0.211	-0.245	-0.059	-0.432	0.003	-0.157	-0.135
Cosinus	-0.008	0.013	-0.013	-0.022	-0.003	-0.005	-0.012	-0.020	0.001	-0.035	0.005
Dice	-0.001	0.003	0.032	0.036	0.022	0.012	0.029	0.002	0.005	-0.021	0.003
Best	0.782	0.765	0.548	0.573	0.623	0.798	0.570	0.695	0.439	0.527	-0.213

Table 1: Analysis of the impact of the system components

6 Conclusion

This paper presents the SATLab participation in SemEval 2024 Task 1: Semantic Textual Relatedness (STR). The proposed system predicts semantic relatedness by means of the Euclidean distance between two sentences, calculated on the basis of the frequency of the ngrams of characters that make them up. It employs no resources external to the material and extracts no information from other instances present in the material. The system performs well, coming first in five of the twelve languages. However, there is little difference between the best systems. What's more, the baseline proposed by the organizers was better than all the systems proposed by the participants in two languages.

Analysis of the system's components shows that the decision to develop a fully instance-specific approach was clearly the wrong one. Simply taking into account the frequencies of features in the material as a whole, as the TF-IDF weighting system does, provides a significant benefit, as Damashek (1995) has already pointed out when character ngrams are used in other NLP tasks.

The performance of all teams varies considerably according to language. It would be very interesting to carry out further research to try and understand the origin of these fluctuations. Otherwise, this type of unsupervised approach cannot be recommended, since negative correlations are observed for one of the languages. It is possible that this is linked to the way in which the material has been designed, which varies greatly depending on the language for obvious reasons of unavailability of certain resources (Ousidhoum et al., 2024a).

7 Ethical Considerations

The ethical issues raised by this research are identical to those described by the researchers who collected the data (Ousidhoum et al., 2024a) and by the researchers who organized this task (Ousidhoum et al., 2024b).

Acknowledgements

The author wishes to thank the organizers of this shared task for putting together this valuable event and for their availability throughout the task. He is a Research Associate of the Fonds de la Recherche Scientifique - FNRS (Fédération Wallonie Bruxelles de Belgique).

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Yves Bestgen. 2020. Boosting a KNN classifier by improving feature extraction for authorship identification of source code. In *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation*, CEUR Workshop Proceedings, pages 705–712. CEUR-WS.org.

Yves Bestgen. 2021a. Optimizing a supervised classifier for a difficult language identification problem. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 96–101, Kiyv, Ukraine. Association for Computational Linguistics.

Yves Bestgen. 2021b. A simple language-agnostic yet strong baseline system for hate speech and offensive content identification. In *Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation*, CEUR Workshop Proceedings, pages 1–10. CEUR-WS.org.

Yves Bestgen. 2022. Please, don't forget the difference and the confidence interval when seeking for the state-of-the-art status. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5956–5962, Marseille, France. European Language Resources Association.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Marc Damashek. 1995. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843–848.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full

sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Fuchun Peng, Dale Schuurmans, and Shaojun Wang. 2003. Language and task independent text categorization with simple language models. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 189–196.

Genaios at SemEval-2024 Task 8: Detecting Machine-Generated Text by Mixing Language Model Probabilistic Features

Areg Mikael Sarvazyan and **José Ángel González** and **Marc Franco-Salvador**
Genaios, Valencia, Spain
{areg.sarvazyan, jose.gonzalez, marc.franco}@genaios.ai

Abstract

This paper describes the participation of the Genaios team in the monolingual track of Subtask A at SemEval-2024 Task 8. Our best system, LLMIXTIC, is a Transformer Encoder that mixes token-level probabilistic features extracted from four `LLaMA-2` models. We obtained the best results in the official ranking (96.88% accuracy), showing a false positive ratio of 4.38% and a false negative ratio of 1.97% on the test set. We further study LLMIXTIC through ablation, probabilistic, and attention analyses, finding that (i) performance improves as more LLMs and probabilistic features are included, (ii) LLMIXTIC puts most attention on the features of the last tokens, (iii) it fails on samples where human text probabilities become consistently higher than for generated text, and (iv) LLMIXTIC's false negatives exhibit a bias towards text with newlines.

1 Introduction

The analysis of Machine-Generated Text (MGT) has gained popularity in recent times. This is important for detecting and attributing text to Large Language Models (LLMs) such as LLaMA (Touvron et al., 2023) and GPT (Ouyang et al., 2022), and combating fake-news, intellectual property violations (Henderson et al., 2023), data leakages (Nasr et al., 2023), among other malicious usages (Kasneci et al., 2023). Recent efforts include zero-shot (Bao et al., 2024) and supervised systems (Wang et al., 2023). However, large-scale scenarios that combine domains, data sources, or models are still challenging (Sarvazyan et al., 2023b; Eloundou et al., 2023). As a result, different frameworks to generate high-quality MGT datasets[1] (Sarvazyan et al., 2024) and evaluation campaigns have been released (Shamardina et al., 2022; Sarvazyan et al., 2023a). In this paper, we describe our solution as the Genaios team at SemEval-2024 Task 8: *Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection* (Wang et al., 2024a).

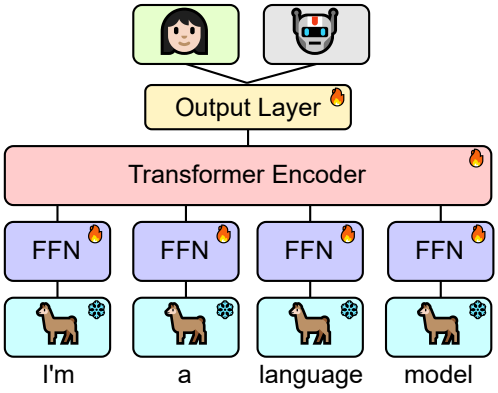


Figure 1: Overview of the proposed system. Modules marked with ❄ are frozen. Those with 🔥 are trainable.

Our starting point is the observation that LLMs assign higher probabilities to MGT than to human text. We propose LLMIXTIC, illustrated in Figure 1, which leverages this via a Transformer encoder (Vaswani et al., 2017) that mixes token-level probabilistic features extracted from four `LLaMA-2` models, both instructed and base flavors: `LLaMA-2-7b`, `LLaMA-2-7b-chat`, `LLaMA-2-13b`, and `LLaMA-2-13b-chat`. For each token, our features are (i) the log probability of the observed token, (ii) the log probability of the predicted token, and (iii) the entropy of the distribution.

These probabilistic features capture MGT style in a precise manner, favouring detection. As a result, we obtained the best results in the official ranking (96.88% accuracy) for the monolingual track of Subtask A: *Binary Human-Written vs. Machine-Generated Text Classification*. Our analysis shows that performance improves as more LLMs and probabilistic features are used. In addition, LLMIXTIC pays more attention to the last tokens of the sequence, where higher probabilities for human texts lead to misclassifications. Finally,

[1] One of these is TextMachina, freely available at `https://github.com/Genaios/TextMachina`

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 101–107

	Label	Model	arXiv	PeerRead	Reddit	WikiHow	Wikipedia		Outfox
Train	👩		15.5	2.4	15.5	15.5	14.5	Test	16.2
	🤖	Bloomz	-	-	-	-	-		3
		Cohere	3	2.3	3	3	2.3		3
		ChatGPT	3	2.3	3	3	3		3
		Davinci	3	2.3	3	3	3		3
		Dolly	3	2.3	3	3	2.7		3
		GPT4	-	-	-	-	-		3
Dev	👩		0.5	0.5	0.5	0.5	0.5		-
	🤖	Bloomz	0.5	0.5	0.5	0.5	0.5		-

Table 1: Statistics of the Subtask A Monolingual dataset by split, label, model, and domain. Sizes in thousands.

texts with newlines are predominant among false negatives.

2 Background

The monolingual track of Subtask A: *Binary Human-Written vs. Machine-Generated Text Classification* focuses on detecting whether an English text is entirely written by a 👩 **human** or generated by an 🤖 **LLM**. The data is an extension of the M4 dataset (Wang et al., 2024b) and combines texts from different domains and LLMs. We show the statistics of the dataset in Table 1. The official evaluation metric of the Subtask A is accuracy, which we also employ in our experiments.

3 System Overview

It is known that high-quality human text does not follow high-probability distributions over the next tokens (Holtzman et al., 2020). In contrast, LLMs are decoded to sample from regions of high probability, thus assigning higher probability to low-diversity constructions and lower to human texts. In practice, this causes MGT to be measurably different from human texts, e.g., showing less idiomatic expressions, scarce and repetitive discourse markers, or strictly complying with canonical orderings of constituents (Simón et al., 2023).

We developed our system by following these previous findings, and considering that most of the current LLMs share two key components which condition the probability distributions they learn: (i) the underlying backbone, namely Transformer decoder, with few architectural changes and (ii) large portions of their training data both for pre-training and instruction tuning. Our system relies on the hypothesis that token-level probabilistic features extracted from an specific set of LLMs can be used to differentiate human texts and MGT from a potentially different set of LLMs, which has been shown to be very effective in existing MGT detectors (Przybyła et al., 2023; Wang et al., 2023).

As depicted in Figure 1, our final system is a Transformer Encoder that mixes token-level probabilistic features extracted from four `LLaMA-2` models (Touvron et al., 2023), including base and instructed versions: `Llama-2-7b`, `Llama-2-7b-chat`, `Llama-2-13b`, and `Llama-2-13b-chat`. Following (Przybyła et al., 2023), we build feature sequences where each token is represented as the concatenation of three probabilistic features extracted from each LLM. Specifically, we employ the following features.

Log probability of the predicted token. Measures the highest probability assigned by θ to the next token as:

$$\alpha_i = \max_{y \in \mathcal{V}} \log p_\theta(y|x_{<i}) \quad (1)$$

Entropy of the distribution. Measures the uncertainty of θ for choosing the next token:

$$\beta_i = -\sum_{y \in \mathcal{V}} p_\theta(y|x_{<i}) \log p_\theta(y|x_{<i}) \quad (2)$$

Log probability of the observed token. Measures how likely is the observed token x_i according to the model θ and the prefix $x_{<i}$ as:

$$\gamma_i = \log p_\theta(x_i|x_{<i}) \quad (3)$$

Given a text $x = [x_1, ..., x_n]$ and a set of LLMs $\mathcal{L} = \{\theta_1, ..., \theta_m\}$, we represent x as a feature sequence $h = [h_1, \ldots, h_n]$ with each h_i denoting the probabilistic features from all the LLMs for the i-th token, $h_i = [\alpha_i^1; \beta_i^1; \gamma_i^1, \ldots, \alpha_i^m; \beta_i^m; \gamma_i^m]$. For instance, our final system uses four LLMs and three features from each one, $h \in \mathbb{R}^{n \times 12}$. Note that the features are extracted per-token, which constrains us to use LLMs with a shared tokenizer.

The feature vectors in h are projected to 128 dimensions through a feed-forward layer, and then mixed with a Transformer encoder of 1 layer and 4 attention heads. The output of the Transformer layer is averaged and a softmax layer is used to compute a probability distribution over the human and generated classes. This classifier on top of the probabilistic features, LLMIXTIC's only trainable component, is comprised of solely 85k parameters, being 0.0002% of the total.

4 Experimentation

We focus on the monolingual track of Subtask A, carrying out comparisons among models and ablations of the best system. For these we employ the original training and validation splits provided by the organizers. In the post-evaluation stage, we analyze the errors of LLMIXTIC in the test set by inspecting the probabilistic features extracted from `LLaMa-2`, the learned attention heads, and text patterns in the misclassified samples.

4.1 Model Comparison

We compare LLMIXTIC with classical and neural models, while also evaluating different LLMs to extract the probabilistic features. All the models in these comparisons are trained and evaluated on the original training and validation splits provided by the shared task organizers.

Classical baselines. We consider a Logistic Regression classifier, using either TF-IDF features with word n-grams ranging from 1 to 3-grams (`LR+TFIDF`), or readability features (`LR+READ`). For these, we employ `scikit-learn` (Pedregosa et al., 2011) and `readability`,[2] training the model with balanced class weights and default parameters.

Neural baselines. We also compare LLMIXTIC with two fully fine-tuned Transformer encoders, `roberta-base` (Liu et al., 2019) and `e5-base` (Wang et al., 2022). These models are trained for four epochs, using the cross-entropy loss, a batch size of 32 samples, and a learning rate of 5e-6.

LLMIXTIC's LLMs. We evaluate LLMIXTIC with probabilistic features from two LLM families, namely `GPT-2` (Radford et al., 2019; Sanh et al., 2019) and `LLaMA-2` (Touvron et al., 2023). For the `GPT-2` family,[3] we include `gpt2`, `gpt2-medium`, and `distillgpt2`. The `LLaMA-2` family is comprised of `LLaMA-2-7b`, `LLaMA-2-7b-chat`, `LLaMA-2-13b`, and `LLaMA-2-13b-chat`. These are trained for ten epochs, with a maximum text length of 512 tokens, a batch size of 32 samples, a learning rate of 1e-3, and the cross-entropy loss.

All neural models are trained with HuggingFace's Trainer (Wolf et al., 2020) in FP16 mode, employing early stopping, with a patience of 3 evaluation steps, on the validation set. The LLMs used for feature extraction are always frozen, with `LLaMA-2` models also being quantized to 8 bits. We implement LLMIXTIC in PyTorch (Paszke et al., 2019), and run all the experiments using a single NVIDIA RTX A6000.

Model	Accuracy (%)
`LR+READ`	42.32
`LR+TFIDF`	61.26
`roberta-base`	80.58
`e5-base`	74.48
LLMIXTIC (w/ `GPT-2`)	67.42
LLMIXTIC (w/ `LLaMA-2`)	**85.98**

Table 2: Model comparison results on the dev set.

Results are presented in Table 2. Here we observe how LLMIXTIC using `LLaMA-2` features outperforms every baseline by large margins, improving upon the best baseline's score by 5 points in accuracy, while having only 0.07% relative training parameters. Notably, all the neural models outperform classical baselines, which suggests that grammatical features, especially those based on readability measures, are not enough to properly discriminate between human-written and generated text. Also, the usage of probabilistic features from `GPT-2` models does not yield good results in comparison to neural baselines and LLMIXTIC with `LLaMA-2` LLMs. This suggests that the scale of the LLM used to extract features could have a large impact on the results. Considering that the `LLaMA-2` family is more similar than `GPT-2` models to the LLMs that generated the text of the dataset, we also hypothesize that using feature extraction LLMs that more closely resemble the LLMs in the dataset can yield better results.

4.2 LLM and Feature Ablations

We study the impact the number of LLMs and probabilistic features have on LLMIXTIC's performance by means of two ablation studies: at LLM and at feature level. These experiments are performed with the same experimental setup: first training with a single LLM or feature, and continually adding the other LLMs or features.

Ablation results are presented in table 3. In LLM ablation we observe improvements as more LLMs are included. Notably, the inclusion of `chat` models provides the largest improvements of up to ten points. Building upon our hypothesis about similarities in architecture, training strategies, and datasets

[2] https://github.com/andreasvc/readability/

[3] Chosen for its success in previous shared tasks (Przybyła et al., 2023) and to test for more efficient feature extractors.

Ablation	Configuration	Accuracy (%)
LLMs	LLaMA-v2-7b	74.90
	+ LLaMA-v2-13b	75.86
	+ LLaMA-v2-7b-chat	78.48
	+ LLaMA-v2-13b-chat	**85.98**
Features	Predicted	79.40
	+ Entropy	83.26
	+ Observed	**85.98**

Table 3: Ablation study over LLMs and features.

of instruction-tuned LLMs, it is expected that most of them, especially the chat models we used, have learned close distributions. Therefore, we consider that this improvement can be explained by the nature of the dataset, where all the generators were instruction tuned. We also note that LLMIXTIC with only non-instructed LLMs achieves similar results to one of the neural baselines, outperforming LLMIXTIC with GPT-2 by a large margin.

Similar to the LLM ablation, feature ablation results improve as more features are included, achieving an increment of more than six points when all the features are used. We observe that LLMIXTIC obtains similar performance to the best neural baseline just using the log probability of the predicted token and outperforms it after adding the entropy of the distribution. Besides, only with one feature, the performance is ten points higher than LLMIXTIC with GPT-2 using all the features.

5 Results

Our official submission is LLMIXTIC with LLaMA-2, trained on the training and validation sets, using the previously described experimental setting. Table 4 presents the results obtained by our system, where it reaches an accuracy of 96.88%, surpassing the other participants' approaches and ranking first. Due to time constraints, we focused our participation on the monolingual track. However, having seen the performance of LLMIXTIC on the test set of the monolingual track, we trained LLMIXTIC under the same setting for the multilingual track in a post-deadline stage (denoted in tables with *). Here, we obtained an accuracy of 89.97%, which would have placed us at 14th position.

6 Analysis

We further analyze the behavior of LLMIXTIC in the test set by examining the probabilistic features extracted from LLaMa-2, the learned attention heads, and patterns in misclassified samples.

Track	Rank	Name	Accuracy (%)
Monolingual	**1**	**Genaios**	**96.88**
	2	USTC-BUPT	96.09
	20	*baseline*	88.46
		(119 more)	
Multilingual	1	USTC-BUPT	95.98
	14*	**Genaios**	**89.97**
	25	*baseline*	80.88
		(44 more)	

Table 4: Final results on the official ranking. Bold denotes our team's placement.

LLMIXTIC fails when human text probabilities become larger than for generated texts. In contrast, LLMIXTIC works better when the generated text probabilities are consistently larger than those from human texts. To illustrate this behavior, Figure 2 shows each LLM's feature averaged both for correct and erroneous predicted samples. Errors occur with unusually high values of α and γ features in the human class, and unusually low values for the generated class. The effect of feature β is also notable, with the margin between human and generated curves being smaller in misclassifications. Additionally, for each class, chat and base models reveal different curves for all three features.

LLMIXTIC pays more attention to the last positions. Figure 3 shows the average of the attention heads across all the samples to illustrate it. This behavior could be the main cause of errors when human text probabilities become consistently larger than those for generated texts in the last positions, as shown in Figure 2. A diagonal pattern with high probability is also noticeable until approximately position 150, after which it disappears.

Human text is more often confused with generated text than vice versa. There are twice as many false positives as there are false negatives (714 vs. 355). This translates into a false positive rate of 4.38% and a false negative rate of 1.97%.

Newlines are predominant in false negatives. We manually analyze the errors with higher confidence, finding that most of LLMIXTIC's false negatives include \n to separate sentences or paragraphs, while false positives do not, to the same extent. Specifically, \n is present in 75.49% of false negatives, whereas it is only present in 34.59% of false positives. This difference could suggest (i) a potential bias in the training data, with human texts containing more \n than the generated texts,

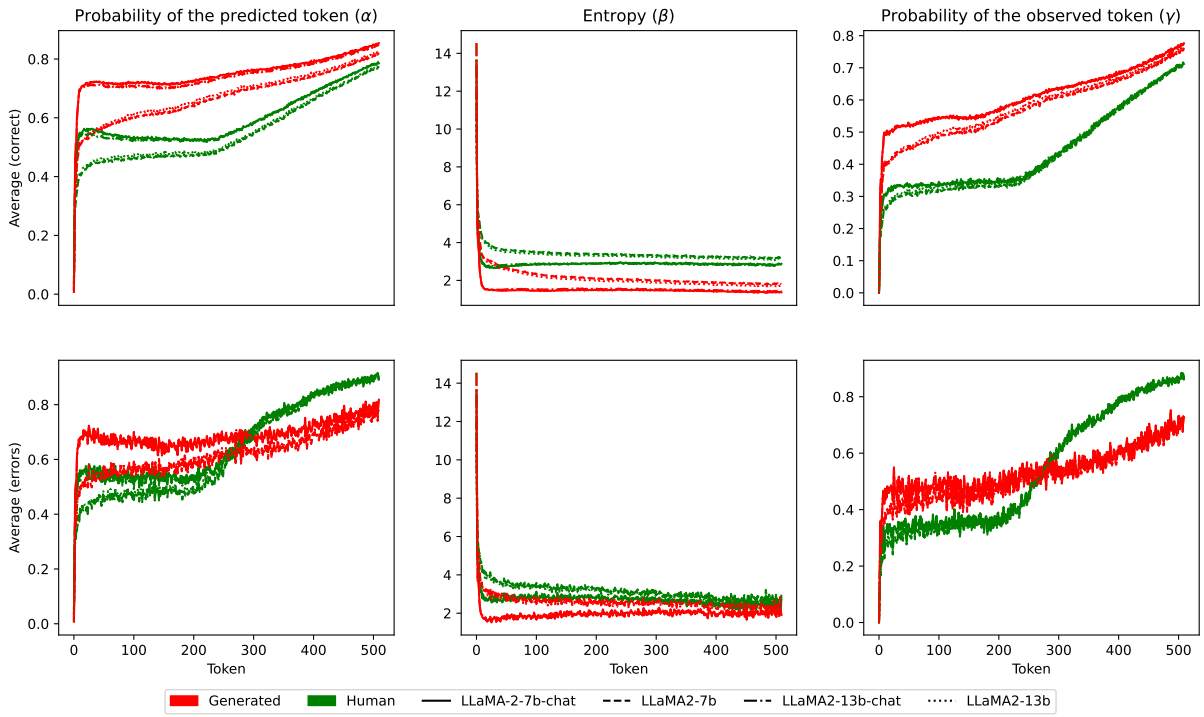


Figure 2: Sample-averaged probabilistic features of the four `LLaMa-2` models, for the two classes (generated and human). Both for correct predictions (top row) and errors (bottom row). The y axis denotes the average of the probabilistic feature (α, β, or γ) across all samples of a label in the test set, at a given position marked on the x axis. Throughout all positions, the probabilities of generated text for correct predictions consistently exceed those of humans. However, for errors, human probabilities surpass those of generated text from the middle of the sequences.

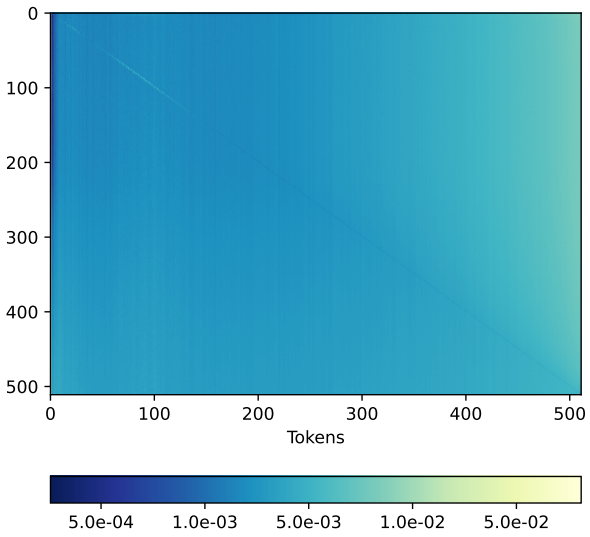


Figure 3: Sample-averaged and head-averaged attention scores from LLMIXTIC's Transformer encoder. LLMIXTIC pays more attention to the last positions.

or (ii) our system is learning a spurious correlation between `\n` and the human class.

7 Conclusion

We described the participation of the Genaios team in the monolingual track of Subtask A at SemEval-2024 Task 8. We proposed LLMIXTIC, a Transformer Encoder that mixes token-level probabilistic features extracted from four base and instructed `LLaMA-2` models, namely `LLaMA-2-7b`, `LLaMA-2-7b-chat`, `LLaMA-2-13b`, and `LLaMA-2-13b-chat`. Our system obtained the best results in the official ranking, with small false positive and false negative ratios.

Our ablation analyses showed that LLMIXTIC's performance improves as more LLMs and probabilistic features are used. We compared these features across correctly predicted and misclassified samples, finding that LLMIXTIC works better when MGT probabilities are consistently higher than for human text. In addition, attentions are mostly focused on the last tokens, which could be one of the causes of the errors made by LLMIXTIC. Finally, the newline character seems predominant in false negatives but not in false positives, which suggests biases either in the data or in our model.

Aiming to foster R&D in this area, future works will focus on TextMachina,[1] a framework to generate MGT datasets for tasks such the ones addressed in this SemEval shared task: detection, attribution, boundary, and mixcase detection.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*.

Peter Henderson, Xuechen Li, Dan Jurafsky, Tatsunori Hashimoto, Mark A Lemley, and Percy Liang. 2023. Foundation models and fair use. *arXiv preprint arXiv:2303.15715*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, page 102274.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Piotr Przybyła, Nicolau Duran-Silva, and Santiago Egea-Gómez. 2023. I've seen things you machines wouldn't believe: Measuring content predictability to identify automatically-generated text. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023). CEUR Workshop Proceedings, CEUR-WS, Jaén, Spain*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*.

Areg Mikael Sarvazyan, José Ángel González, and Marc Franco-Salvador. 2024. TextMachina: Seamless Generation of Machine-Generated Text Datasets. *arXiv preprint arXiv:2401.03946*.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023a. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains. *Sociedad Española de Procesamiento del Languaje Natural (SEPLN)*, 71:275–288.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, and Paolo Rosso. 2023b. Supervised machine-generated text detectors: Family and scale matters. In *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization*. Springer International Publishing.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the the ruatd shared task 2022 on artificial text detection in russian. *arXiv preprint arXiv:2206.01583*.

Lara Alonso Simón, José Antonio Gonzalo Gimeno, Ana María Fernández-Pampillón Cesteros, Marianela Fernández Trinidad, and María Victoria Escandell Vidal. 2023. Using linguistic knowledge for automated text identification. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023). CEUR Workshop Proceedings, CEUR-WS, Jaén, Spain*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156, Singapore. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Self-StrAE at SemEval-2024 Task 1: Making Self-Structuring AutoEncoders Learn More With Less

Mattia Opper [a] and **N. Siddharth** [a,b]
[a] University of Edinburgh; [b] The Alan Turing Institute
{m.opper,n.siddharth}@ed.ac.uk

Abstract

This paper presents two simple improvements to the Self-Structuring AutoEncoder (Self-StrAE). Firstly, we show that including reconstruction to the vocabulary as an auxiliary objective improves representation quality. Secondly, we demonstrate that increasing the number of independent channels leads to significant improvements in embedding quality, while simultaneously reducing the number of parameters. Surprisingly, we demonstrate that this trend can be followed to the extreme, even to point of reducing the total number of non-embedding parameters to seven. Our system can be pre-trained from scratch with as little as 10M tokens of input data, and proves effective across English, Spanish and Afrikaans.

1 Introduction

Natural language is generally understood to be compositional. To understand a sentence, all you need to know are the meanings of the words and how they fit together. The mode of combination is generally conceived as an explicitly structured hierarchical process which can be described through, for example, a parse tree. Recent work by Opper et al. (2023) presents the Self-StrAE (Self-Structuring AutoEncoder), a model which learns embeddings such that they define their own hierarchical structure and extend to multiple levels (i.e. from the subword to the sentence level and beyond). The strengths of this model lie in its parameter and data efficiency achieved through the inductive bias towards hierarchy.

Learning embeddings such that they meaningfully represent semantics is crucial for many modern NLP applications. For example, retrieval augmented generation (Lewis et al., 2020) is predicated on the fact that the correct contexts for a given query can be determined. The semantic relation between a query and a context is encompassed by the notion of semantic relatedness. They are not equivalent to one another (i.e. paraphrases), but are close in meaning in a broader, more contextual sense. The focus of task one of this year's SemEval (Ousidhoum et al., 2024a,b) is capturing this notion of semantic relatedness, with a particular focus on African and Asian languages generally characterised by a lack of NLP resources.

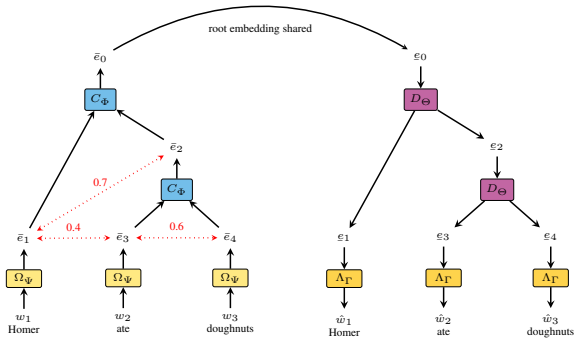


Figure 1: Self-StrAE forward pass. Red lines indicate cosine similarity between adjacent nodes. Shared colours indicate shared parameters.

In this work, we investigate whether Self-StrAE can learn embeddings which capture semantic relatedness, when trained from scratch on moderately sized pre-training corpora. We turn to the competition in order to examine whether the model can even compare with dedicated STR systems. In order to determine whether Self-StrAE can provide an alternative approach in low resource settings where systems that rely on large pre-trained transformers (Vaswani et al., 2017) may not have sufficient scale to prove effective. We show that with two simple changes, Self-StrAE's performance can be substantially improved. Moreover, we demonstrate that the the resulting system is not limited to English, but can work equally well (if not better) for both Spanish and Afrikaans [1].

[1] Code available at: https://github.com/mopper97/Self-StrAE

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 108–115

2 Model and Objectives

2.1 Model

The core architecture at the heart of this paper is the Self-StrAE. A model that processes a given sentence to generate both multi-level embeddings and a structure over the input. The forward pass begins by first *embedding* tokens to form an initial frontier, using the embedding matrix Ω_Ψ. This is followed by iterative application of the following update rule:

1. Take the cosine similarity between adjacent embeddings in the frontier.
2. Pop the most similar pair.
3. Merge the pair into a single parent representation, and insert into the frontier.
4. If len(frontier) = 1, stop

Merge is handled by the recursively applied *composition function* C_Φ, which takes the embeddings of two children and produces that of the parent. The process is illustrated in 1. In the figure, the highest cosine similarity is between the embeddings of 'ate' and 'doughnuts', so these two embeddings are merged first. At the next step, 'Homer' and 'ate doughnuts' are merged as they have the highest similarity of the remaining embeddings. At this point the frontier has shrunk to a single embedding and the root has been reached.

If we consider the merge history at the root, we can see that it has come to define a tree structure over the input. This structure is passed to the decoder, which then generates a second set of embeddings, starting from the root and proceeding to the leaves. The decoder achieves this through recursive application of the *decomposition function* D_Θ, which takes the embedding of a parent and produces the embeddings of the two children. Once the decoder reaches the leaves, it can optionally output discrete tokens through use of a *dembedding function* Λ_Γ.

We denote embeddings produced during composition as $\bar{e}$ and produced during decomposition as $\underline{e}$. For a vocabulary of size *V*, each embedding $e \in \mathbb{R}^E$ consists of *k* independent channels of size *u*. With this notation established, we can now define the four core components of a Self-StrAE.

Embedding:
$\Omega_\Psi(w_i) = w_i\Psi$, where $\Psi \in \mathbb{R}^{V\times E}$

Composition:
$C_\Phi(\bar{e}_{c1}, \bar{e}_{c2}) = hcat(\bar{e}_{c1}, \bar{e}_{c2})\Phi + \phi$
where $\Phi \in \mathbb{R}^{2u\times u}$ and $\phi \in \mathbb{R}^u$

Decomposition:
$D_\Theta(\underline{e}_p) = hsplit(\underline{e}_p\Theta + \theta)$
where $\Theta \in \mathbb{R}^{u\times 2u}$ and $\theta \in \mathbb{R}^{2u}$

Dembedding:
$\Lambda_\Gamma(\underline{e}_i) = \underline{e}_i\Gamma$ where $\Gamma \in \mathbb{R}^{E\times V}$

Note that in the above the dembedding layer is treated as a separate parameter matrix to the embedding layer, however, it can just as easily be weight tied to increase efficiency.

2.2 Objectives

There are a few options for pre-training Self-StrAE. The simplest solution is to have the model reconstruct the leaf tokens, which can be achieved by simply employing cross entropy over the output of the dembedding layer. For a given sentence $s_j = \langle w_i \rangle_{i=1}^{T_j}$, this objective is formulated as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{T_j}\sum_{i=1}^{T_j} w_i \cdot \log \hat{w}_i. \tag{1}$$

An alternative approach adopted by Opper et al. (2023) is to use contrastive loss as the reconstruction objective. For a given batch of sentences s_j, the total number of nodes (internal + leaves) in the associated structure is denoted as M. This allows for the construction of a pairwise similarity matrix $A \in \mathbb{R}^{M\times M}$ between normalised upward embeddings $\langle \bar{e}_i \rangle_{i=1}^M$ and normalised downward embeddings $\langle \underline{e}_i \rangle_{i=1}^M$, using the cosine similarity metric (where embeddings are flattened to be of shape *E*). Denoting $A_{i\bullet}, A_{\bullet j}, A_{ij}$ the i^{th} row, j^{th} column, and $(i,j)^{\text{th}}$ entry of a matrix respectively, the objective is defined as:

$$\mathcal{L}_{\text{cont}} = \frac{-1}{2M}\left[\sum_{i=1}^{M}\log \sigma_\tau(A_{i\bullet}) + \sum_{j=1}^{M}\log \sigma_\tau(A_{\bullet j})\right] \tag{2}$$

where $\sigma_\tau(\cdot)$ is the tempered `softmax` (temperature τ), normalising over the unspecified ($\bullet$) dimension.

A final option is to combine these two objectives, applying the cross entropy reconstruction over leaves and the contrastive objective over all other nodes, where constructing a vocabulary is intractable due to the number of possible combinations. The contrastive objective remains identical except that A is now defined as pairwise similarity matrix $A \in \mathbb{R}^{I \times I}$, where I is the number of internal nodes of the structure. In its simplest form, this objective, which we will henceforth refer to as CECO, can then be defined as:

$$\mathcal{L}_{\text{CECO}} = \frac{1}{2}(\mathcal{L}_{CE} + \mathcal{L}_{cont}) \quad (3)$$

3 Experiments

3.1 Setup

For all experiments, we utilise a pre-training set of ≈10 million tokens. We make this choice because Self-StrAE is intended to be data efficient, especially if it is to be useful for low resource languages where scale may not be available. For English the data was sourced from a subset of Wikipedia, while for Afrikaans and Spanish we obtained corpora from Leipzig Corpora Collection[2]. We utilise a pre-trained BPE tokenizer for each language from the BPEMB Python package (Heinzerling and Strube, 2018). Though the package also provides pre-trained embeddings, we solely use the tokenizer and learn embeddings from scratch.

During the course of model development, we utilised additional evaluation sets as a further guide. For English, we used Simlex (Hill et al., 2015) and Wordsim353 (Agirre et al., 2009) as measures of how well the model captures lexical semantics, and STS-12 (Agirre et al., 2012), STS-16 (Agirre et al., 2016) and STS-B (Cer et al., 2017). For Afrikaans, due to lack of resources, we utilised a Dutch translation of STS-B (Huertas-García et al., 2021) as the two languages are closely related. For Spanish, we utilised a Spanish translation of STS-B from the same source, as well as the labelled train and dev sets from SemRel 2024 (Ousidhoum et al., 2024a). While these sets contain labels, we apply the model fully unsupervised and solely use them for zeroshot evaluation.

We train Self-StrAE for 15 epochs using the Adam optimizer at a learning rate of 1e-3 (Kingma and Ba, 2015). We set the embedding dimension to 256, with a batch size of 512 and τ of 1.2. We conducted our primary experiments on English and then applied the same system design to Spanish and Afrikaans.

[2]For both Spanish and Afrikaans we selected the mixed corpus and took a uniform subsample to reduce size to the requisite scale.

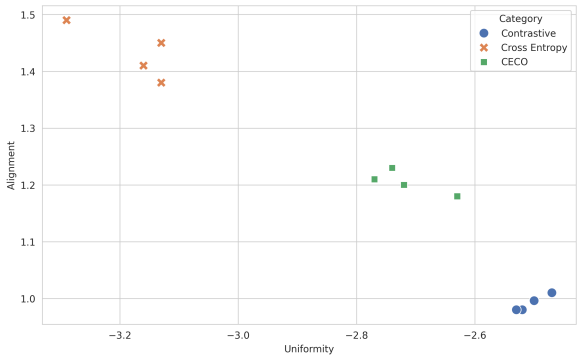


Figure 2: Uniformity and Alignment plot for contrastive, cross entropy and CECO pre-training objectives. Results taken across four random seeds. Lower is better for both measures.

3.2 Which Objective is Best?

The first thing we want to establish is which objective is most suitable for training Self-StrAE, as the original version only utilises contrastive loss. For parity with the original implementation, we treat the embeddings as square matrices (i.e. $k = u$) in this experiment.

Figure 2 show the uniformity and alignment analysis (Wang and Isola, 2020) of the representations learned by the different objectives. Uniformity describes the extent to which embeddings are spread around the space, while alignment characterises how similar positive target pairs are to each other. To be successful, representations should optimise both properties. We can observe that while the cross entropy objective leads to uniformity, it is comparatively poor at optimising alignment. This essentially implies that the decoder embeddings deviate from those of the encoder. Alignment is clearly a desireable property, as the results in table 1 show. The contrastive loss leads to both better sentence level representations and to more stable performance.

However, the best setting of all is CECO (the combination of cross entropy and contrastive). There are two factors worth considering that may explain this finding. Firstly, including reconstruction of discrete labels inherently provides additional meaningful information compared to just organising the representations alone. Secondly, at the token level the contrastive loss is most sus-

Objective	Simlex	Wordsim S	Wordsim R	STS-12	STS-16	STS-B	SemRel (Dev)
Contrastive	13.80 ± 0.41	54.33 ± 0.78	52.40 ± 0.87	31.93 ± 1.03	52.48 ± 0.44	40.05 ± 2.01	50.13 ± 0.88
CE	13.77 ± 9.43	46.43 ± 24.00	51.23 ± 23.04	17.68 ± 4.88	25.40 ± 15.60	22.43 ± 15.12	32.95 ± 14.93
CECO	**19.15 ± 2.39**	**58.33 ± 3.31**	**62.65 ± 2.76**	**41.20 ± 4.04**	**58.40 ± 1.35**	**48.35 ± 1.36**	**54.40 ± 0.81**

Table 1: Comparison of Objective Performance. Results are taken across four random intialisations. Models are trained on English.

k	u	Simlex	Wordsim S	Wordsim R	STS-12	STS-16	STS-B	SemRel (Dev)	# Params
8	32	17.50 ± 2.12	58.45 ± 1.04	62.10 ± 2.29	31.00 ± 2.67	52.53 ± 3.33	41.90 ± 2.09	49.30 ± 0.59	4192
32	8	17.28 ± 5.94	44.83 ± 27.11	49.10 ± 25.47	33.28 ± 17.49	46.75 ± 30.85	41.35 ± 25.57	43.95 ± 30.50	280
64	4	16.15 ± 9.82	48.63 ± 20.95	51.30 ± 23.05	38.88 ± 22.39	49.48 ± 31.05	43.05 ± 28.91	46.13 ± 30.35	88
128	2	17.33 ± 7.12	52.85 ± 19.33	55.15 ± 19.85	39.63 ± 20.83	50.38 ± 31.92	46.63 ± 27.95	47.78 ± 30.92	22
256	1	12.00 ± 12.84	42.80 ± 23.35	45.05 ± 24.58	29.18 ± 24.68	39.65 ± 32.22	37.35 ± 29.55	40.63 ± 29.07	7
8	32	19.4	59.4	64.3	27.6	56	44.5	50.1	4192
32	8	21.6	57.2	61.6	44.3	63.3	54.1	58.8	280
64	4	**21.7**	62.8	66.1	49.9	65.6	57.4	61.3	88
128	2	18.4	**65.1**	**67.2**	49	**67.2**	60.9	63.2	22
256	1	20.7	63.2	66.3	**50.1**	66.2	**61.6**	**63.6**	7

Table 2: Impact of number of independent channels on performance. Results are taken across four random initialisations. Models are trained on English. Top half of the table represents average performance, the bottom half contains the best performing initialisation. # Params is the number of non-embedding parameters.

ceptible to noise (e.g. the word 'the' may occur frequently in the batch, but each repeated instance will be treated as a false negative), and under such conditions the objective has been shown to lead to feature suppression (Robinson et al., 2021).

Summary: We find that combining cross entropy and contrastive loss leads to better representations than applying each objective individually, and consequently use this approach going forward.

3.3 How many channels?

Each embedding in Self-StrAE is treated as consisting of k independent channels of size u. This is intended to allow the representations to capture different senses of meaning. However, in the original paper the number of channels is set to be the square root of u, and not explored further. Consequently, we wanted to see what the optimal balance between the number of channels and their size was. Results are shown in 2. Surprisingly, we found that as the number of channels increased (and consequently u decreased) performance improves quite dramatically, even to the limit of treating each value in the embedding as independent. Furthermore, because the number of non-embedding parameters (i.e. the composition and decomposition functions) is directly tied to the channel size u, *decreasing model complexity improves embedding quality*.

However, it should be noted that this decrease in complexity comes with a tradeoff in terms of reliability. The smaller the size of the channel, the more variance we observed between random initialisations, with some initialisations failing to learn any meaningful representations whatsoever. We have found a solution that is able to maintain performance and ensure stability between seeds, but we leave discussion of this to the appendix, as we do not yet have a clear picture of what exactly is causing instability and wish to avoid speculation. We do however wish to emphasise that the problem is tractable and there is ample scope for further development, and direct the interested reader to A for more information.

Summary: Increasing the number of channels while decreasing their size leads to significant improvements in performance, though at the cost of some instability between seeds. For our submisson to SemRel we used the setting $k = 128$, $u = 2$ as this allowed for an acceptable failure rate while not compromising performance (roughly 1 in 4 seeds fail). Consequently, our system utilises only 22 non-embedding parameters.

3.4 Performance Across Languages

So far our experiments have only considered English. We now examine whether the framework is language agonistic, and pre-train Self-StrAE on both Spanish and Afrikaans. As before we pretrain on a small scale data (described in 3.1).

Results are in 3. We can see that the improvements to Self-StrAE hold across different languages and are not the result of some quirk in our English pre-training set. In fact performance is either comparable or better than on English. The

Language	NL STS-B (Dev)	NL STS-B (Test)	Afr SemRel (Dev)	Afr SemRel (Test)	Competition Rank
Afrikaans	52.8	64.5	23.4	76.5	2
Language	ESP STS-B (Test)	ESP SemRel (Train)	ESP SemRel (Dev)	ESP SemRel (Test)	Competition Rank
Spanish	61.5	58.5	68.7	63.5	6

Table 3: Self-StrAE Performance on Spanish and Afrikaans. Results correspond to those of the submitted systems, which we selected using the best run from four random initialisations.

results on Afrikaans are particularly interesting as the model performs significantly better on this language. Whether this is due to how the test set was created or to underlying features of the language provides an interesting question for future work. Moreover, the Afrikaans model, despite never having been trained on Dutch, is able to generalise fairly well to it, shown by the results on the translated STS-B sets.

4 Related Work

Recursive Neural Networks: Self-StrAE belongs to the class of recursive neural networks first popularised by (Socher et al., 2011, 2013). Recursive neural networks are extremely similar to recurrent neural networks, they differ because they process inputs hierarchically rather than sequentially (e.g. going up a parse tree).

Learning Structure and Representations: Recursive neural networks require structure as input. An alternative approach is to train a model that learns structure and the network at the same time. Recent unsupervised examples include Drozdov et al. (2019, 2020); Hu et al. (2021). However, these mechanisms generally use search to determine structure making them highly memory intensive. Self-StrAE differs from these as it asks the representations to define their own structure, making it much more resource efficient, though less flexible in certain aspects.

Contrastive Loss: Contrastive loss is an objective which optimises the representation space directly. In broad terms this objective requires the representations of a positive pair to be as similar to each other as possible, while minimising similarity to a set of negative examples. The closest examples of this objective, for the approach employed in this paper, are Chen et al. (2020); Shi et al. (2020); Radford et al. (2021).

5 Conclusion

We show that two simple changes can make Self-StrAE significantly more performant: adding a discrete reconstruction objective and increasing the number of independent channels. The latter also has the added benefit of reducing the number of parameters in the model, and surprisingly means that simpler is better. More broadly, we believe these findings demonstrate the potential of an inductive biases towards explicit structure. Self-StrAE, at present, is a very simple model. The only thing it really has going for it is the inductive bias which tasks embeddings with organising themselves hierarchically. While the gap between Self-StrAE and SoTA systems still remains, the fact that it is able to perform at all demonstrates the promise. Moreover, the fact that the two simple changes demonstrated in this paper can lead to such improvements indicates that the full potential of the inductive bias has yet to be reached, and it is likely that further refinements can lead to even more substantial benefits. Finally, because this model does not require significant scale to optimise pursuing further improvements may provide substantial benefits for low resource languages where pre-training data is scarce.

6 Limitations

The results in this paper represent steps towards an improved model rather than a complete picture. We still do not fully understand what causes the instability in training when the number of channels increased, and though we can provide a solution (see A), further analysis is needed. The performance of contrastive loss can depend quite heavily on how positive and negative examples are defined and it is likely that the explanation rests there. Secondly, while we have shown that Self-StrAE can be applied to languages other than English the results are limited to Indo-European languages. An interesting avenue for future work would be investigating a broader spectrum of languages, and whether specific characteristics can be identified which influence how well the model performs.

7 Acknowledgements

MO was funded by a PhD studentship through Huawei-Edinburgh Research Lab Project 10410153. We thank Victor Prokhorov, Ivan Vegner and Vivek Iyer for their valuable comments and helpful suggestions during the creation of this work.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 19–27.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, page 385–393, USA. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Workshop on Semantic Evaluation (SemEval)*, pages 1–14.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Learning Representations (ICLR)*, volume 119, pages 1597–1607.

Andrew Drozdov, Subendhu Rongali, Yi-Pei Chen, Tim O'Gorman, Mohit Iyyer, and Andrew McCallum. 2020. Unsupervised parsing with S-DIORA: Single tree encoding for deep inside-outside recursive autoencoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4832–4845, Online. Association for Computational Linguistics.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1129–1141, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Xiang Hu, Haitao Mi, Zujie Wen, Yafang Wang, Yi Su, Jing Zheng, and Gerard de Melo. 2021. R2D2: Recursive transformer based on differentiable tree for interpretable hierarchical language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4897–4908, Online. Association for Computational Linguistics.

Álvaro Huertas-García, Javier Huertas-Tato, Alejandro Martín, and David Camacho. 2021. Countering misinformation through semantic-aware multilingual models. In *Intelligent Data Engineering and Automated Learning – IDEAL 2021*, pages 312–323, Cham. Springer International Publishing.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Mattia Opper, Victor Prokhorov, and Siddharth N. 2023. StrAE: Autoencoding for pre-trained embeddings using explicit structure. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7544–7560, Singapore. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said

Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. 2021. Can contrastive learning avoid shortcut solutions? *CoRR*, abs/2106.11230.

Yuge Shi, Brooks Paige, Philip Torr, and N. Siddharth. 2020. Relating by contrasting: A data-efficient framework for multimodal generative models. In *International Conference on Learning Representations (ICLR)*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–161.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *CoRR*, abs/2005.10242.

A Stabilising High Channel Self-StrAE

One solution we have found to the instability issue is modifying the objective. This formulation, loosely inspired by SimCSE (Gao et al., 2021), runs the same input through the model twice, with different dropout masks applied each time. The objective is cross entropy reconstruction for the leaves, and contrastive loss between the two different sets of decoder embeddings for the non-terminals. Currently we have two theories as to why this might work:

- Better negatives: because the decoder embeddings represent the contextualised meaning of node rather than it's local one, the issue of false negatives is somewhat mitigated.
- Encoder consistency: because we ask the two sets of decoder embeddings to be similar to each other the encoder is encouraged to produce the same structure regardless of dropout mask. It may be that this pressure towards regularity leads to the improved consistency.

Results are shown in 4. For lack of a better term we refer to this alternative objective as StrCSE. In its current form we do not consider this objective to be well formed, and solely provide it here as a possible starting point for further research.

Objective	Simlex	Wordsim S	Wordsim R	STS-12	STS-16	STS-B	SemRel (Dev)
Contrastive	13.80 ± 0.41	54.33 ± 0.78	52.40 ± 0.87	31.93 ± 1.03	52.48 ± 0.44	40.05 ± 2.01	50.13 ± 0.88
CE	13.77 ± 9.43	46.43 ± 24.00	51.23 ± 23.04	17.68 ± 4.88	25.40 ± 15.60	22.43 ± 15.12	32.95 ± 14.93
CECO	**19.15 ± 2.39**	**58.33 ± 3.31**	**62.65 ± 2.76**	41.20 ± 4.04	58.40 ± 1.35	48.35 ± 1.36	54.40 ± 0.81
CECO k=128 u=2	17.33 ± 7.12	52.85 ± 19.33	55.15 ± 19.85	39.63 ± 20.83	50.38 ± 31.92	46.63 ± 27.95	47.78 ± 30.92
StrCSE k=128 u=2	**21.68 ± 1.88**	**59.06 ± 2.38**	**64.08 ± 0.91**	**49.46 ± 0.59**	**66.18 ± 0.24**	**61.30 ± 0.76**	**62.88 ± 0.42**

Table 4: StrCSE compared with other objectives. Results are taken over four random intialisations. Training data is English.

RGAT at SemEval-2024 Task 2: Biomedical Natural Language Inference using Graph Attention Network

Abir Chakraborty
Microsoft
Abir.Chakraborty@microsoft.com

Abstract

In this work, we (team RGAT) describe our approaches for the SemEval 2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials (NLI4CT). The objective of this task is multi-evidence natural language inference based on different sections of clinical trial reports. We have explored various approaches, (a) dependency tree of the input query as additional features in a Graph Attention Network (GAT) along with the token and parts-of-speech features, (b) sequence-to-sequence approach using various models and synthetic data and finally, (c) in-context learning using large language models (LLMs) like GPT-4. Amongst these three approaches the best result is obtained from the LLM with 0.76 F1-score (the highest being 0.78), 0.86 in faithfulness and 0.74 in consistence.

1 Introduction

Clinical trials are advanced treatments and tests to evaluate new ways of treating life-threatening diseases where interventions include new drugs, cells and other biological products, advanced surgical or radiological procedures and devices. As the trial progresses the observations are documented systematically in a Clinical Trial report that includes the subject selection criteria ('Eligibility'), treatments ('Interventions') and results at group level including adverse effects. These reports constitute a rich source of past endeavours to learn from and help in formulating new treatment plans. However, the sheer volume of CT reports[1] makes it impossible to conduct extensive manual evaluation. Thus, it is necessary to have an automated pipeline that can enquire a CT report for specific hypothesis and provides high accuracy and reliability at the same time.

[1]As of Jan 17, 2024, ClinicalTrials.gov lists 480,795 CT studies

Natural language inference or NLI (Devlin et al., 2019) is one of the standard NLP tasks where a hypothesis is qualified as true (entailment) or false (contradiction) or even undetermined (neutral) given a premise. This task is adopted for reasoning over CT reports by Jullien et al. (2023) where two new tasks are created based on NLI4CT dataset, (1) NLI over CT reports and (2) extracting the evidence/mention from CT reports to support the inference label. The Semeval 2024 Task 2 NLI4CT is also based on the same NLI4CT dataset (identical for training) with modifications in the test split (more details in the Data section). The inferencing is challenging as it requires multi-hop reasoning, i.e., dependency and aggregation are required over different pieces of the document.

Other than the complexity associated with multi-hop reasoning, the domain and the associated word-distribution also creates significant challenge due to the presence of aliases, acronyms and biomedical terminologies (Lee et al., 2019a; Shickel et al., 2018; Jin et al., 2019). This results in significant drop in model performance as is evident in the NLI results last year (Jullien et al., 2023) where it was found that majority of the submitted solutions failed to outperform the baseline solution with a significant margin. The challenge is also evident in the overall performance of models on general NLI datasets (e.g., Stanford NLI or SNLI) where the best model results in 93.1% F1-score (Wang et al., 2021).

When it comes to different modeling approaches, many of the top-performing models for the SNLI dataset are ensemble in nature. While initial individual models are based on RNN, most of the latest ones are based on the Transformer architecture and pretrained language models like RoBERTa or T5. Similar trend can also be seen in Jullien et al. (2023) where the best model is an ensemble and both DeBERTa and Flan-T5 made their way to the top. Interestingly, LLMs like GPT3.5 could not

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 116–122

make a significant boost in the performance.

In our approach, we explored three different modeling paradigms, namely, (1) custom Graph Attention Network (GAT) based discriminative model with novel features based on the dependency tree of the input query, (2) generative models based on T5 and Flan-T5 but enriched with synthetic data used for both pre-training and fine-tuning, and (3) LLM like GPT-4 applied with and without few-shot examples. It is not surprising that the best performance was obtained by GPT-4 stressing on the importance of generic knowledge (that is embedded in these LLMs) rather than fine-tuning, especially when the dataset is not large enough.

The organization of the paper is as follows. In the next section we provide a detailed literature survey on the techniques employed for NLI. Next, we present the details of the proposed approaches. Subsequently, the model predictions and comparisons with other baseline methods are discussed. Finally, conclusions are drawn and scope for future works is outlined.

2 Related Work

The existing body of work for the general NLI is quite rich where they are based on the Stanford NLI (SNLI) dataset (550k examples but restricted to a single text genre) (Bowman et al., 2015) and three other NLI datasets present in GLUE (Wang et al., 2018), namely, MNLI, QNLI and WNLI. The MNLI (Multi-Genre Natural Language Inference Corpus) dataset (Williams et al., 2018) is a crowd-sourced NLI dataset gathered from different sources, e.g., government reports (and covers different genres, e.g., fiction, travel). Given a premise-hypothesis pair of sentences, the task is to predict one of the three classes, namely, whether the premise sentence entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral). The QNLI is modified from Stanford Question Answer Dataset (Rajpurkar et al., 2016) where the task is to determine whether the context sentence contains the answer to the question. Similarly, the WNLI dataset is created from the Winograd Schema Challenge (Levesque et al., 2012) where a coreference resolution problem is converted into an entailment problem involving a pronoun and its referent. Another large NLI dataset is multi-genre NLI (MNLI) that has 433k examples covering multiple genres and supporting cross-genre evaluation. Some of the best performances are obtained by RoBERTa (Liu et al., 2019b), XLNet (Yang et al., 2020), Multi-Task Deep Neural Network (MT-DNN) (Liu et al., 2019a) and generative pre-training (GPT) approach (Radford et al., 2018).

There are few NLI datasets in the biomedical domain, namely, MedNLI (Romanov and Shivade, 2018) and BioNLI (Bastan et al., 2022). MedNLI has 14k example pairs created by clinicians on 4,683 premises with three categories, entailment, contradiction and neutral. BioNLI, on the other hand, goes beyond sentence-level inference and includes large context as premises that requires handling complex texts as well as domain knowledge. Bastan et al. also includes negative examples as adversarial hypothesis using nine strategies which is a speciality of this dataset.

There are three biomedical domain specific models that are typically used on these datasets. Starting with the available weights of BERT (pretrained on general domain corpora), BioBERT (Lee et al., 2019b) is trained on PubMed abstracts and PMC full-text articles and shown to outperform BERT on NER, relation extraction and Q&A, all in the biomedical domain. PubMedBERT (Gu et al., 2021) is a BERT model created from scratch (rather than starting with general domain corpora) on large biomedical domain dataset like PubMed and achieved impressive performance for tasks like NER and Q&A. BioLinkBERT (Yasunaga et al., 2022) further exploited links between PubMed documents to create a richer context that is used to build a language model (LM). This model has obtained SOTA performance on biomedical datasets such as BLURB (Gu et al., 2021) and BioASQ (Nentidis et al., 2020). Another model that achieved SOTA performance on MedNLI is SciFive (Phan et al., 2021) which is based on T5 paradigm.

There are not many studies on the application of Graph Neural Network for NLI. Inspired by KIM (Chen et al., 2018) where external knowledge is infused for NLI task, Song et al. (2020) developed a joint training model where Graph Attention Network (GAT) is used to represent the sub-graph associated with entities that are involved in the hypothesis. Another closely related GAT application is from Chen et al. (2021) applied for fact verification on Wikipedia articles. Typical applications of GAT in the NLP domain are for question answering, semantic parsing, information extraction and Named Entity Recognition (Wu et al., 2022;

Chakraborty, 2023).

3 Task Description & Data

The dataset for Multi-evidence NLI for Clinical Trial (NLI4CT) is based on a collection of breast-cancer CT reports[2] containing statements, explanations and labels annotated by domain expert annotators (Jullien et al., 2024). Each CT report has four sections: (a) Eligibility criteria (a set of conditions for patients to be included in the trial cohort), (b) Intervention (information regarding the details of treatments administered), (c) Results (what is the outcome of these treatments) and (d) Adverse events (if anything was observed during the period of the trial). The annotated statements (hypothesis) are claims extracted from one of the four sections (with an average length of 19.5 tokens) and may even compare more than one report. Each statement is qualified as either 'Contradiction' or 'Entailment'.

There are 1700 examples in the training set and 200 in the development/validation set with exactly 50:50 split of the two classes. The test set has 5500 examples with unknown label distribution. A typical example looks like the following:

1. **Hypothesis**: 'All the primary trial participants do not receive any oral capecitabine, oral lapatinib ditosylate or cixutumumab IV, in contrast all the secondary trial subjects receive these.'
2. **Primary context**: 'Patients with early stage, ER positive primary breast cancer undergo FLT PET scan at baseline and 1-6 weeks after the start of standard endocrine treatment. The surgery follows 1-7 days after the second FLT PET scan.'
3. **Secondary context**: 'Patients receive oral capecitabine twice daily on days 1-14 and oral lapatinib ditosylate once daily on days 1-21. Courses repeat every 21 days in the absence of disease progression or unacceptable toxicity'
4. **Label**: 'Contradiction'

where the secondary context provides the justification of the label.

4 Methodology

We have explored three different modeling strategies for the prediction of the inference label. They are (1) custom discriminative model with GAT applied to create features from the dependency tree of the hypothesis statement, (2) sequence-to-sequence generative models based on T5 and Flan-T5 but enriched with synthetic data used in both pre-training and fine-tuning and (3) LLM based solution with and without Few-shot examples.

[2]extracted from https://clinicaltrials.gov/ct2/home

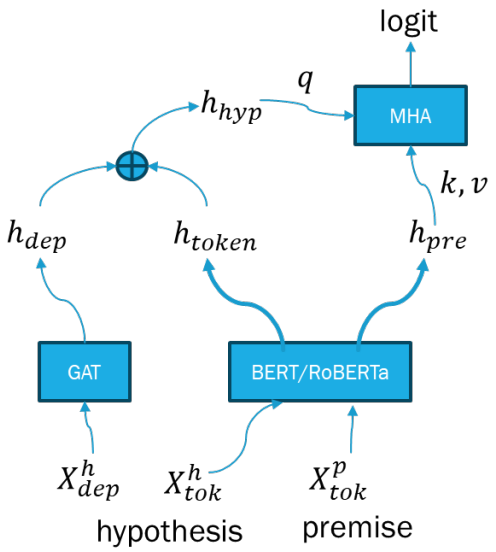


Figure 1: The architecture of the custom model using GAT and Multi-head attention (MHA).

4.1 Discriminative Model

The architecture of our custom discriminative model is shown in Fig 1. We use the tokens of both the hypothesis and the premises to generate a representation using either a standard BERT or RoBERTa model (referred as h_{tokens} for the hypothesis and h_{pre} for the premise. Following the RGAT approach of Wang et al. (2020) (originally meant for aspect polarity detection) we utilize the dependency structure of the input hypothesis (X^h_{dep}) that captures the grammatical relations by connecting the words with the corresponding dependency type. However, we do not reorient the dependency tree since there is no aspect word in our application. Using GAT based processing of the hypothesis dependency tree we generate additional features h_{dep}. Details of the GAT based processing are provided in Appendix A. We concatenate both the features of the hypothesis (h_{dep} and h_{tokens}) and pass through a linear layer to create the final hypothesis feature, h_{hyp}. For the premise, there is only the token based feature, h_{pre}, which is used as a key and value in a standard multi-head attention (MHA) with h_{hyp} as the query vector. This process

is repeated multiple times (maximum 3) with the output of the previous MHA layer. Finally, we take the first vector of the MHA output (corresponding to [CLS]) and pass it through a linear layer to generate the logits. The model is trained for binary cross-entropy loss.

4.2 Generative Model

In the 2023 SemEval challenge (Jullien et al., 2023), it was found that generative models outperformed discriminative models on the entailment task. We also explore different T5 models (small and base T5 and base SciFive) for the current entailment task with the exception that we have also generated synthetic data for pre-training as well as fine-tuning.

4.2.1 Generation of Synthetic Data

For generating synthetic data for T5 pre-training we follow (1) the standard T5 random span masking[3] for both the hypothesis and premise sentences and (2) ask GPT-4 to identify spans and mask them subsequently. The first approach works better for the quality of the data and we use this approach for generating the final pre-training data. We have used noise density = 0.4 and average noise span length of 2 and generate 73,457 pre-training examples.

For generating additional fine-tuning data, we use GPT-4 (with temperature = 0.7) with three additional tasks, namely, (a) Question answering on the premise text, and (b) additional inference data from the same set of premises and (c) create a contradictory hypothesis from the original hypothesis. For the first task, examples look like

1. Question: 'How many weeks after the start of standard endocrine treatment is the second FLT PET scan conducted?', Answer: '1-6 weeks'

2. Question: 'On which days is oral capecitabine given in Arm A?', Answer: 'days 1-14'

Additional NLI examples are

1. Hypothesis:No adverse events were reported in the clinical trial., Label: Entailment

2. Hypothesis: The clinical trial report had 765 adverse events in one section and 88 in another section., Label: Contradiction

In this process we generate 11k Q&A pairs and 45k NLI pairs and 1700 contradictory NLI examples from the original 1700 training examples.

4.3 Large Language Model

It was also observed in 2023 SemEval challenge (Jullien et al., 2023) that increase in model size also improves the performance. We further validate this hypothesis by applying GPT-4 to the NLI task with and without few-shot examples.

4.4 Implementation Details

For the discriminative model we use the bi-affine parser (Dozat and Manning, 2016) from AllenNLP for dependency parsing. For all experiments, the embedding dimension for the dependency relation is same as the hidden dimension of the BERT/RoBERTa model. We use 3 MHA layers with 8 heads and 2 GAT layers with 6 heads and all the dropouts are fixed at 0.3. The model has a total of 110 million parameters for BERT-base and 351 million parameters for BERT-large. The last hidden state of the pre-trained BERT[4] is used for the initial token representations which is subsequently fine-tuned. All models are trained for 50 epochs using Adam optimizer (Kingma and Ba, 2014) (with the default parameters), a learning rate of 5×10^{-5} and a batch size of 8.

We have pretrained both small and base T5 models for subsequent NLI task. Pretraining is done for 20 epochs with a batch size of 16 and learning rate of 5×10^{-5} with Adam optimizer. From the 73,457 span masked examples, we use 66111 for training and 7346 for validation that is used to keep track of the validation loss and saving the model.

5 Results

In this section, first we describe the performance of the custom discriminative model followed by the performance of the fine-tuned T5 model and finally the results from GPT-4. Although we compute precision, recall and F1-score for all our experiments we report only F1-score here. It is to be noted that we did not evaluate our model on the test dataset for all our experiments and submitted test results only for the best validation performance. Thus, for most of our experiments we report only the validation F1-score and also mention the test F1-score wherever available. Table 1 summarizes the results from the custom discriminative model. There are

[3]https://github.com/google-research/text-to-text-transfer-transformer

[4]https://github.com/huggingface/transformers

Model Type	Base Model	Model Parameters	Dev-F1	Test-F1
Cross-attention	BERT-base	110 M	0.64	
Combined pooler	BERT-base	110 M	0.65	
Cross-attention + GAT	BERT-base	110 M	**0.67**	0.49
Cross-attention + GAT	BERT-large	351 M	**0.67**	0.50

Table 1: Performance of the custom discriminative model on the validation and test dataset

Model Type	Model	Additional Data	Dev-F1	Test-F1
random initial weight	small T5 (60.5 M)	None	0.55	
random initial weight	small T5 (60.5 M)	synthetic NLI data-I	0.51	
random initial weight	small T5 (60.5 M)	synthetic NLI data-II	0.53	
pretrained with CTR data	small Flan-T5 (76 M)	None	0.58	
pretrained	base T5 (223 M)	None	0.64	
pretrained	base T5 (223 M)	Synthetic Q&A data	0.43	
pretrained	base T5 (223 M)	Synthetic NLI-I data	0.55	
pretrained	base T5 (223 M)	Synthetic NLI-II data	0.54	
pretrained	Flan-T5 base (247 M)	None	0.66	0.608
pretrained	Flan-T5 base (247 M)	Synthetic NLI-I	-	0.535
-	GPT-4 (`0613`)	Zero-shot	-	**0.761**

Table 2: Performance of different generative models including GPT-4.

four flavors of this model, one with BERT-large and three with BERT-base. Within BERT-base, we have one with cross-attention, one without ('combined-pooler' that only concatenates the two BERT outputs) and the third one with cross-attention and GAT. It can be seen that the presence of GAT improves the validation F1 score over the other variants. However, the performance does not improve with the larger BERT model. Surprisingly, the corresponding test F1-score shows significant degradation implying substantial difference in the test data distribution (tokens, nature of problem or label) from that of the validation dataset. The small number of validation dataset also contributes to this mismatch.

Table 2 captures the details of different experiments with generative models like, T5, Flan-T5 and GPT-4. The size of the generative model (small vs. base) has strong contribution to the performance as confirmed earlier (Jullien et al., 2023). However, the addition of synthetic data does not improve (rather degrade) the F1-score which is evident for both the small and base version of T5. This challenges the traditional belief of improvement due to multi-task learning and indicates potential conflicts in the synthetic data due to either a mismatch in the nature of the problem (e.g., Q&A) or accuracy of the synthetic data (since they are not manually verified). The best result is obtained by a base Flan-T5 model trained without any synthetic dataset that results in a test F1-score of 0.61. Finally, using GPT-4 (version `0613`, maximum context length of 8192) without any Few-shot examples results in the best test F1-score of 0.76.

6 Conclusion

In this work we have explored both discriminative and generative models for NLI applied to CT reports. While our custom discriminative model outperforms generative models like T5-base and Flan-T5-base the same is not true when evaluated on the test dataset indicating the limitation of the small validation dataset and significant change in data distribution. Since the training dataset is small (1700) we also explore enriching the same with synthetic data created by LLMs like GPT-4 for additional task (e.g., Q&A) and the same NLI task. However, the addition of these synthetic data substantially degrades the performance rather than improving pointing to a deeper analysis of the role of synthetic data for NLI task. The only exception is in the pretraining synthetic data created for small Flan-T5 model that boosted the final performance. The best result is obtained by GPT-4 without using Few-shot examples and we suspect both the addition of examples and modification of the prompt can further improve the performance.

References

Mohaddeseh Bastan, Mihai Surdeanu, and Niranjan Balasubramanian. 2022. BioNLI: Generating a biomedical NLI dataset using lexico-semantic constraints for adversarial examples. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5093–5104, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Abir Chakraborty. 2023. RGAT at SemEval-2023 task 2: Named entity recognition using graph attention network. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 163–170, Toronto, Canada. Association for Computational Linguistics.

Chonghao Chen, Jianming Zheng, and Honghui Chen. 2021. Knowledge-enhanced graph attention network for fact verification. *Mathematics*, 9(16).

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Qiao Jin, Jinling Liu, and Xinghua Lu. 2019. Deep contextualized biomedical abbreviation expansion. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 88–96, Florence, Italy. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019a. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019b. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Improving multi-task deep neural networks via knowledge distillation for natural language understanding.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Anastasios Nentidis, Konstantinos Bougiatiotis, Anastasia Krithara, and Georgios Paliouras. 2020. *Results of the Seventh Edition of the BioASQ Challenge*, page 553–568. Springer International Publishing.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium. Association for Computational Linguistics.

Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. 2018. Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5):1589–1604.

Meina Song, Wen Zhao, and E. HaiHong. 2020. Kganet: a knowledge graph attention network for enhancing natural language inference. *Neural Comput. Appl.*, 32(18):14963–14973.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. 2022. Graph neural networks for natural language processing: A survey.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

A Graph Attention Network

The dependency tree can be represented by a graph structure where each node is a word and the edges between them are represented by the dependency relation, e.g., nominal subject, adverbial modifier, etc. Following Wang et al. (2020), given a neighborhood of a node $\mathcal{N}_i$, the node embeddings can be iteratively updated using multi-head attention (with K attentional heads) as

$$h_{att_i}^{l+1} = concat_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{lk} W_k^l h_j^l, \quad (1)$$

$$\alpha_{ij}^{lk} = attention(i, j), \quad (2)$$

where $h_{att_i}^{l+1}$ is the attention head of node-i at layer $l+1$ and α_{ij}^{lk} is the normalized attention coefficient computed by the k-th attention at layer l and W_k^l is an input transformation matrix.

In addition to the attention head of word-i a relational head is also computed for this node as

$$h_{rel_i}^{l+1} = concat_{m=1}^{M} \sum_{j \in \mathcal{N}_i} \beta_{ij}^{lm} W_m^l h_j^l, \quad (3)$$

$$g_{ij}^{lm} = \sigma(relu\left(r_{ij} W_{m1} + b_{m1}\right) W_{m2} + b_{m2}) \quad (4)$$

$$\beta_{ij}^{lm} = exp(g_{ij}^{lm}) / \sum_{j \in \mathcal{N}_i} exp(g_{ij}^{lm}) \quad (5)$$

where r_{ij} denotes the relation embedding between node-i and j and M is the number of relational heads. The final representation of each word (node) is a concatenation of the attention and relational embeddings:

$$x_i^{l+1} = concat(h_{att_i}^{l+1}, h_{rel_i}^{l+1}) \quad (6)$$

$$h_i^{l+1} = relu\left(W_{l+1} x_i^{l+1} + b_{l+1}\right) \quad (7)$$

BDA at SemEval-2024 Task 4: Detection of Persuasion in Memes Across Languages with Ensemble Learning and External Knowledge

Victoria Sherratt, Sedat Dogan, Ifeoluwa Wuraola,
Lydia Bryan-Smith, Oyinkansola Onwuchekwa and **Nina Dethlefs**
University of Hull
Big Data Analytics Research Group
v.sherratt-2020@hull.ac.uk

Abstract

This paper outlines our multimodal ensemble learning system for identifying persuasion techniques in memes. We contribute an approach which utilises the novel inclusion of consistent named visual entities extracted using Google Vision's API as an external knowledge source, joined to our multimodal ensemble via late fusion. As well as detailing our experiments in ensemble combinations, fusion methods and data augmentation, we explore the impact of including external data and summarise post-evaluation improvements to our architecture based on analysis of the task results.

1 Introduction

In this paper, we describe our approach to identifying persuasion techniques for SemEval 2024 Task 4. The task involves the identification of up to 22 persuasion techniques in memes, which are inherently multimodal. We participated in Subtask2a and Subtask2b.

Subtask2a is a multilabel classification task, requiring the identification of 22 persuasion techniques using both textual and visual content. The subtask is evaluated by a hierarchical F1, as each label is part of a subset of techniques and contains a parent node. Subtask2b is a binary classification task, determining the presence or absence of any persuasion technique within a meme (propagandistic or non-propagandistic). For both subtasks, training data is provided in the English language and a development set also in English. As well as English, 3 surprise languages in Arabic, North Macedonian and Bulgarian were provided to officially evaluate our approach (Dimitrov et al., 2024).

Our system architecture is an amalgamation of traditional NLP and vision models, exploring late and early fusion techniques as well as carefully crafted confidence thresholds. We extend beyond the training data by incorporating resources such as Google Vision[1], which provides consistent named visual entities extracted from the image regardless of language; in a multilingual context this reduces reliance on sentence spans or tokens, which can be problematic due to linguistic variations in unseen language data. We also make our code publicly available.[2]

2 Background

Identifying persuasion techniques in memes is necessary endeavour for combating misinformation and fostering critical media consumption among the public, and the focus of a number of ongoing research areas for the prevention of harmful content, propaganda or disinformation spread through memes (Dimitrov et al., 2021a; Dupuis and Williams, 2019; Sharma et al., 2022).

Propaganda is generally referred to as information which is purposefully shaped or presented to support a particular agenda, often utilising the persuasion techniques in this shared task. Previous shared tasks have also considered the identification of persuasion techniques in text only (Da San Martino et al., 2020), multimodal contexts using memes (Dimitrov et al., 2021b), and persuasion techniques in multilingual text (Piskorski et al., 2023b). SemEval 2024 Task 4 is a shared task of a similar nature, however the task considers both image and text as well as multilingual test data.

As meaning is often generating through the interaction of both modalities in memes, meme related tasks are typically approached using pretrained convolutional neural networks (Beskow et al., 2020; Hossain et al., 2022; Sherratt et al., 2023; Suryawanshi et al., 2020) or vision transformers (Afridi et al., 2021; Cao et al., 2023) in combination with language models. Our ensemble approach therefore explores CNNs for the binary classifica-

[1] https://cloud.google.com/vision/docs/detecting-web
[2] https://github.com/vemchance/BDA-SemEval4

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 123–132

tion task; for the more complex multilabel classification, we explore CLIP (Radford et al., 2021) to leverage its significant pretraining on large-scale natural language descriptions and images, as well as its notable performance in zero-shot classification and related downstream multimodal tasks such as social media sentiment analysis (Bryan-Smith et al., 2023).

Our motivation for including external knowledge sources is inspired by previous successful applications of external information (Zhu, 2020) and ongoing research to improve meme-related tasks with the addition of structured knowledge to provide context to memes (Sherratt, 2022; Tommasini et al., 2023).

3 Exploratory Analysis

We briefly explore the task data and use this analysis to inform our approach, particularly for the more challenging Subtask2a. Exploring Subtask2a, we calculated TF-IDF vectors for texts within each label and calculated the cosine similarity between these vectors. We noted that, for the majority of labels, there is significant crossover in textual content. We also examine the number of labels in a single meme, as Subtask2a was a multilabel classification problem where each meme could have more than one persuasion technique, in Figure 1.

Given this crossover, we initially explored leveraging the annotation guidelines for the task, which provides concrete examples of how to label each persuasion technique. We noted the annotation guidelines primarily provided examples annotation based on the location of nouns or adjectives per technique, but provided few examples of non-European languages aside from Russian. However, the guidelines did note the presence of 'personal characteristics, organisations, political orientation or opinions' in some techniques (Piskorski et al., 2023a).

We therefore explore a more concise representation of these attributes using the Google Vision API to extract 'web entities' and visual concepts from an image. For multilingual data, this allows us to rely less on sentence spans or tokens - elements that vary across language - and instead leverage visual entities that could consistently represent information for each label regardless of textual content. In Table 1, we outline a sample of extracted entities from Google Vision's web entities search.

Technique	Entity	Occurrence Count
Appeal to (Strong) Emotions	Russia	48
Appeal to (Strong) Emotions	United States	35
Appeal to (Strong) Emotions	Amnesty International	34
Doubt	Brand	52
Doubt	Politics	48
Doubt	Public Relations	40
Doubt	Speech	39
Red Herring	Entrepreneur	8
Red Herring	Business	7
Red Herring	Ukraine	7
Red Herring	Russia	7

Table 1: Example Entities Extracted via Google Vision

4 System Overview

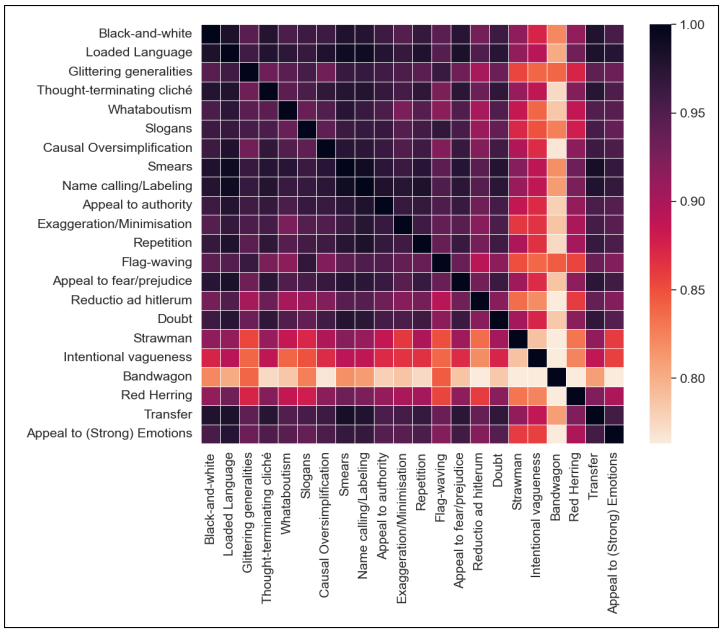
Our main system approach includes ensembling NLP models with vision models for both subtasks. We experimented with BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) family models as well as VGG19 (Simonyan and Zisserman, 2014), ResNet50 (He et al., 2015) and CLIP (Radford et al., 2021).

For Subtask2a, we initially design an architecture that combines multilingual text processing with visual analysis. Our vision stream also includes web entities from Google Vision, processed by a single BERT model. Our Subtask2b system similarly integrates visual and textual modalities with experiments in late and early fusion. We also include additional novel implementations beyond an ensemble of pretrained models:

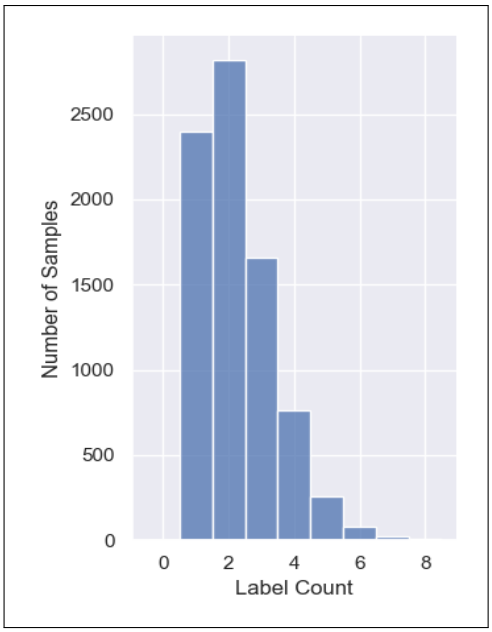
External Knowledge: We use Google Vision to extract information from meme images. The Google Vision API annotates an image using web detection, returning a list of predicted labels for objects, people or concepts in an image, as well as matching URLs and the Google Knowledge Graph ID (Singhal, 2012). We utilise only the named visual entities, with an example in Table 1.

Data Augmentation: We experiment with augmenting the task data. English training data is direct translated using GPT-3.5 (Brown et al., 2020) into a number of other languages, and then again translated when the test datasets are released.

F1 Confidence Threshold: For Subtask2a, we leverage the provided hierarchy of techniques (Dimitrov et al., 2024) to change the confidence threshold for predicted labels. The F1 Confidence Threshold reduces both the threshold required to classify a label from 0.50 to 0.40 (a full reward when scored) and a confidence between 0.35 and 0.40 will return the parent node of the label (partial reward when scored). We detail the impact of the F1 Confidence Threshold in Section 5.2.



(a) TF-IDF Cosine Similarity in Label Groups



(b) Count of Labels Per Meme in Subtask2a

Figure 1: Subtask2a Multilabel Classification Label Exploration

Late Fusion Engine: We implement a late fusion system to combine our separate NLP and vision streams together into a single predictive value. We calculate the per-label accuracy for each model, and use this to weight the contribution of each. In other words:

$$predict_{label} = \frac{(A_{label} \times accA_{label}) + (B_{label} \times accB_{label})}{accA_{label} + accB_{label}}$$

where $accA_{label} \in \{0..1\}$ and $accB_{label} \in \{0..1\}$ refers to the accuracy for the respective models for a given label.

5 Experimental Setup

We combine the training and validation sets for Subtask2a and Subtask2b to train each architecture, a total of 7,500 for Subtask2a and 1,350 for Subtask2b originally in English. We test our approach on the Development Set in English (1,000 samples for Subtask2a and 300 for Subtask2b). Detailed in Section 5.1, the total samples are increased by direct translating data for both subtasks. For all experiments, we set the validation split in the model to 30% of the total training data. When multiple languages are included in the data, we stratify the training and test splits based on language.

The number of epochs is determined by no improvement to validation loss after 5 epochs. We find that the majority of the language models in combination complete around 8 - 10 epochs, whereas CLIP often stops improving around 6 epochs. Table 2 details the specific parameters of our main models. We use pretrained models for both image and text modalities, and therefore the drop-out rate is applied before the respective classification layer detailed in Figure 2.

	mBERT	XLM-RBase	BERT	CLIP
Optimizer	AdamW	AdamW	AdamW	Adam
Dropout	0.4	0.4	0.3	0.5
Weight Decay	1e-5	1e-5	-	-
Learning Rate	1e-5	1e-5	1e-5	5e-5
Batch Size	8	8	8	16

Table 2: Model Parameters

5.1 Additional Data

We explore the use of the Persuasion Techniques Corpus (PTC) (Da San Martino et al., 2020) as additional training data. We use the Google Vision API to extract descriptive entities for all task data images, which is returned in English from the API under the 'web entities' search response. We also augment our dataset using GPT-3.5 (Brown et al., 2020) to direct translate a sample of 500 texts from Subtask2a for each unseen language in the task (1,500 additional samples, or 20% of the available training data). We perform the same process for Subtask2b. Notably we do not augment or change the image for this additional data.

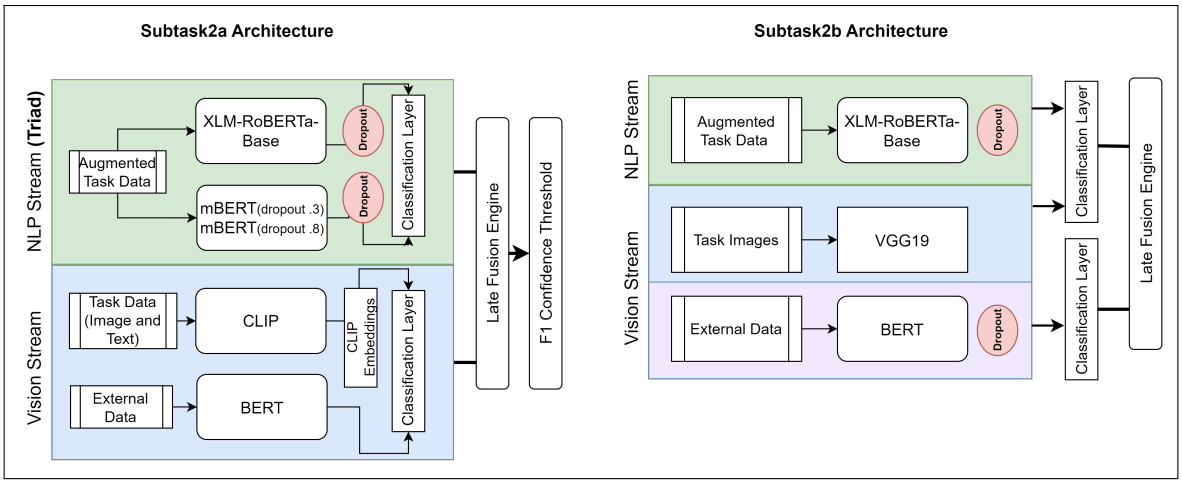


Figure 2: Subtask2a and Subtask2b Architecture

In our results detailed in Section 6, we refer to the Persuasion Techniques Corpus as *PTC*, the original task data as *TD*, the task data with added samples as *ATD* (augmented task data) and data extracted via Google Vision as *ED* (External Data). When external data is used as input, this is followed by (ex) (e.g., BERT(ex)) in Section 6.

5.2 Subtask2a Details

For Subtask2a, we experiment with a number of individual and ensemble models as detailed in Section 6, as well as different fusion strategies and the inclusion of the F1 Confidence Threshold. In early fusion, models are jointly trained and their learned feature vectors concatenated before passed through final classification layer. In late fusion, we use the late fusion engine detailed in Section 4 on the predicted probabilities of each model.

The original architecture is detailed in Figure 2. The three-model NLP stream is referred to the 'Triad' model in experiments, which includes an additional mBERT model with high drop-out to combat over-fitting. However, as we experimented with a number of model combinations, input data and fusion techniques, we opted to choose the model which performed the best on the English development data for the official submission.

As detailed in Table 3 in Section 6, our original architecture was less effective than other experiments. In our final submitted architecture we remove CLIP, so only the BERT model with external data as input remains in the vision stream, and use late fusion to merge this with the Triad NLP architecture. This model is referred to as Traid + BERT(ex) in Table 3.

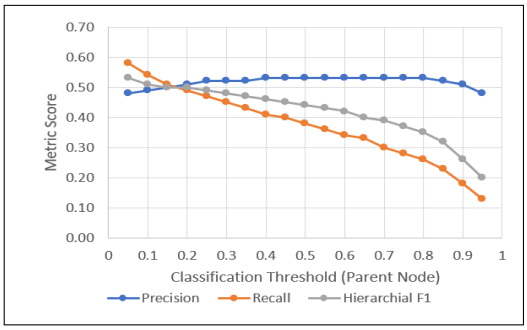


Figure 3: F1 Score Against Parent Node Threshold

We also examine the impact of changing the required confidence threshold for a label, testing a single mBERT model from our ensemble. Figure 3 provides an example each metric score mapped against the threshold to return a parent node label. The F1 Confidence Threshold reduces the threshold required predict a technique, and then introduces another lower threshold to predict the technique label's parent node from the task hierarchy (Dimitrov et al., 2024). We opted to use a configuration which balances the Hierarchical F1, Precision and Recall. In the F1 Hierarchy Threshold, the parent node prediction is always 0.05 less than the label confidence threshold. The configuration used is 0.40 for the label threshold, and 0.35 to return the parent node of the label.

5.3 Subtask2b Details

For Subtask2b, if a model is reused from Subtask2a (e.g., BERT(ex) models to process external data) we reuse the parameters described above. For the vision models, we use a different learning rate for ResNet50 and VGG19 with the AdamW optimizer

Model	Fusion	Finetune Data	H. F1	Precision	Recall
XLM-RBase	-	PTC	0.213	0.362	0.151
XLM-RBase	-	PTC, ATD	0.387	0.516	0.310
XLM-RBase	-	ATD	0.404	0.521	0.330
mBERT	-	PTC	0.213	0.362	0.151
mBERT	-	PTC, ATD	0.163	0.512	0.097
mBERT	-	ATD	0.463	0.523	0.416
BERT(ex)	-	ED	0.395	0.528	0.316
BERT(ex)^{F1}	-	ED	0.424	0.477	0.382
CLIP	-	TD	0.315	0.375	0.272
CLIP^{F1}	-	TD	0.405	0.413	0.398
mBERT + XLM-RBase	Early	ATD	0.451	0.514	0.402
mBERT + XLM-RBase ^{F1}	Early	ATD	0.480	0.471	0.490
mBERT + XLM-RBase + BERT(ex)^{F1}	Early	ATD, ED	0.475	0.466	0.484
CLIP + BERT(ex)	Early	ATD, ED	0.342	0.374	0.316
CLIP + BERT(ex)	Late	ATD, ED	0.345	0.523	0.257
CLIP + BERT(ex)^{F1}	Early	ATD, ED	0.457	0.420	**0.501**
CLIP + BERT(ex)^{F1}	Late	ATD, ED	0.435	0.488	0.392
Triad	Early	ATD	0.470	0.515	0.433
Triad + BERT(ex)	Early	ATD, ED	0.473	0.467	0.480
Triad + BERT(ex)	Late	ATD, ED	0.476	0.470	0.484
Triad + BERT(ex)^{F1}	Late	ATD, ED	**0.483**	0.526	0.446
Triad + BERT(ex) + CLIP	Late	TD, ATD, ED	0.463	**0.541**	0.405
Triad + BERT(ex) + CLIP^{F1}	Late	TD, ATD, ED	0.455	0.461	0.450

Table 3: Subtask2a Experiment Results on Development Set (English)

of 1e-8, a batch size of 8 and the same early stopping parameters as Subtask2a.

Both image models utilise ImageNet weights (Deng et al., 2009). We apply the same dropout rate specified in Table 2 to the text model before this is passed through a classification layer in the case of early fusion. As Subtask2b is a binary classification task, we do not require the F1 Confidence Threshold for this architecture. In our final architecture, VGG19 and XLM-RoBERTa-Base are trained jointly on the augmented task data, and the late fusion engine combines predictions from from the Google Vision web entities.

6 Development Set Results

We detail the results of our experiments for Subtask2a in Table 3 and Subtask2b in Table 4. In the Table 3, the F1 Confidence Threshold modification is indicated by [Model] *F1*.

For Subtask2a, we found the Triad combination performed best with BERT (trained on the extracted Google Vision entities, model BERT(ex) in Table 3) predictions combined with late fusion. The F1 Hierarchy threshold increased the score of the same model in the majority of cases.

Whilst we explored the use of PTC to finetune our models, we found that, due to the different naming conventions of some techniques, performance did not improve with incorporation of the PTC data. We also noted the PTC data was drawn from a different domain (e.g., news articles) were the context of techniques would be longer than short sentences in memes, and potentially this corpus was less effective as a finetuning dataset for the task.

We originally aimed to leverage CLIP's text and image embeddings to inform a novel early fusion neural network model for multilabel multiclass persuasion techniques classification. However, this architecture including CLIP was slightly less effective than others. The reasons behind this suboptimal performance could be multifaceted, including the complexity and subtlety of propagandistic content within memes, the inherent challenges of cross-modal understanding in this particular domain. One reason is suggested that, whilst the visual modality is important for identifying whether

Model	Fusion	Data	F1 Macro	F1 Micro
BERT(ex)	-	ED	0.577	0.580
CLIP	-	TD	0.618	0.680
CLIP + BERT(ex)	Late	TD, ED	0.634	0.707
Triad	Early	ATD	0.383	0.613
VGG19 + BERT	Early	ATD	**0.753**	**0.806**
VGG19 + mBERT	Early	ATD	0.621	0.740
ResNet50 + mBERT	Early	ATD	0.638	0.700
VGG19 + XLM-RBase	Early	ATD	0.641	0.706
ResNet50 + XLM-RBase	Early	ATD	0.618	0.706
VGG19 + XLM-RBase + BERT(ex)	Early	ATD, ED	0.337	0.360
VGG19 + XLM-RBase + BERT(ex)	Late	ATD, ED	*0.677*	0.717
VGG19 + XLM-RBase + CLIP + BERT(ex)	Late	TD, ATD, ED	0.602	0.707

Table 4: Subtask2b Experiment Results on Development Set (English)

a technique is present, *distinguishing* between the specific types of techniques may primarily be a linguistic task.

For Subtask2b, our architecture achieved overall better scores than Subtask2a. We tested architectures retrained for a binary classification task from Subtask2a on Subtask2b as a comparison, noting these models did not perform as well. In Subtask2b, therefore, the vision modality was significant in the binary classification task. We note from the results monolingual language models outperform multilingual models, and suggest this may be due to the limited sample size for the augmented data in Subtask2b. In line with our system strategy, we include BERT(ex) only in conjunction with multilingual models, as the aim of this additional data is to improve zero-shot classification irrespective of language. We observed significant performance increase using the BERT(ex) model in late fusion for Subtask2b.

7 Test Set Performance and Analysis

For the test set, we submitted the best performing model from each subtask experiment. For Subtask2a, this was the Triad + BERT(ex) with late fusion. For Subtask2b, we submitted the VGG19 + BERT model for English test sets and the VGG19 + XLM-RoBERTa-Base + BERT(ex) for all other languages.

Evaluating our results on the test set in Table 5, we found that our model for Subtask2a generalised better on different languages, outperforming the results on the English Development dataset in some cases. Our system performed the best on North Macedonian and the worst in Arabic for this task. The original and augmented task data for Subtask2a was larger than Subtask2b, and we effectively traded English language performance for better generalisability on other languages.

	Rank	F1	Baseline (*Diff.*)
Subtask2a			
English	12	0.504	0.447 (+*0.057*)
Bulgarian	6	0.483	0.500 (-*0.017*)
North Macedonian	**5**	**0.514**	0.555 (-*0.041*)
Arabic	7	0.416	0.486 (-*0.070*)
Subtask2b			
English	**6**	**0.793**	0.250 (+*0.543*)
Bulgarian	9	0.506	0.167 (+*0.339*)
North Macedonian	11	0.435	0.091 (+*0.344*)
Arabic	9	0.510	0.227 (+*0.283*)

Table 5: Results on Official Test Set Leaderboard

For Subtask2b, our architecture under-performed from tests on the English Development dataset aside from the VGG19+BERT model used in the English test set. This approach was less able to generalise on non-English data than our approach from Subtask2a, with a significant score reduction in North Macedonian, our highest scoring language for Subtask2a.

7.1 Subtask2a Test Set Results Analysis

We examine the importance of each modality using the English Development set using the late fusion engine, which calculates the per accuracy label from each model. Table 6 shows the weights of our original architecture (Triad plus CLIP) alongside visual entities extracted from Google, including only the top entity categories with the highest occurrence count.

Technique	NLP Weight	Vision Weight	Top Entities (English)
Appeal to (Strong) Emotions	0.793	0.949	Amnesty International; United States; Product; Russia
Appeal to authority	0.831	0.932	Quotation; US President; United States; Public Relations
Appeal to fear/prejudice	0.916	0.920	Russia; US President; United States; Product
Bandwagon	0.902	0.982	US Vice President; Product; United States; US President
Black-and-white Fallacy/Dictatorship	0.881	0.896	Russia; US President; United States; Product
Causal Oversimplification	0.921	0.943	Public; United States; Public Relations; Product
Doubt	0.912	0.944	Public speaking; Speech; Public Relations; Product
Exaggeration/Minimisation	0.868	0.927	Product; United States; US President
Flag-waving	0.847	0.897	Flag; Product; US President; United States; Speech
Glittering generalities (Virtue)	0.690	0.907	Product; Public Relations; United States; US President
Loaded Language	0.694	0.747	US President; Public Relations; United States; Product
Misrepresentation of Someone's Position (Straw Man)	0.817	0.989	Humor; Russia; US President; United States
Name calling/Labeling	0.648	0.743	Public Relations; US President; United States; Product
Obfuscation, Intentional vagueness, Confusion	0.988	0.988	2023; Album cover; Getty Images; Product
Presenting Irrelevant Data (Red Herring)	0.990	0.990	Business; Ukraine; Russia; Entrepreneur
Reductio ad hitlerum	0.984	0.984	Al-Qaeda; Russia; Product; United States
Repetition	0.961	0.951	Public Relations; Politics; US President; Product; United States
Slogans	0.905	0.883	Public Relations; US President; United States; Product
Smears	0.645	0.468	United States; US President; Product; Public Relations
Thought-terminating cliché	0.906	0.486	Russia; Politics; United States; Product
Transfer	0.733	0.718	Ukraine; United States; Russia; Product
Whataboutism	0.942	0.818	Public Relations; US President; Presentation; Product

Table 6: NLP and vision stream weighting with corresponding visual entities (Subtask2a English Development set)

In Table 6 both streams have a high and sometimes equal weight. Examining the entities, we see that higher weights in the vision stream sometimes corresponds to an identifiable and obvious visual entity - for example, 'Straw Man' or 'Name Calling' techniques with a slightly higher weight for the visual stream are labels which are likely to require a target that may not be present in the text; the top entities for these types of meme usually include a US President or Russia in the English Development set.

Techniques where the weighting leans towards the NLP stream include abstract entities; public relations is often the most common entity before a named entity such as a 'US President' or 'Product'. Additionally, techniques that use linguistic techniques (such as 'Repetition' or 'Slogans', 'Whataboutism', 'Thought-terminating cliché) had a higher contribution from the NLP stream.

7.2 Subtask2b Test Set Results Analysis

For Subtask2b, we noted that the visual modality performed better than models re-trained from Subtask2a. We also noted that, whilst CLIP performed well, as with Subtask2a this was not the best performing visual model. We suggest that VGG19's ability to capture complex visual features were more relevant to the dataset in comparison to CLIP's generalised image-text representations.

Our approach for Subtask2b did not generalise well in comparison to Subtask2a. Whilst the performance drop could equally be attributed to a smaller augmented data sample in Subtask2b, we also examine North Macedonian memes to understand the reduction of performance on this set.

Language	Entity	Occurrence Count
English	Politics	68
English	United States	62
English	US President	38
Bulgarian	Product	24
Bulgarian	Bulgaria	17
Bulgarian	Public Relations	14
North Macedonian	Cartoon	78
North Macedonian	Public Relations	38
North Macedonian	Poster	28
Arabic	Product	29
Arabic	Humor	12
Arabic	Laughter	11

Table 7: Sample Web Entities for Test Dataset in Subtask2b

Visually, North Macedonian memes were different from memes in other languages, particularly in English; they included a significant number of 'cartoon' type memes and comic strips compared to others, which is also reflected in a sample of visual entities outlined in Table 7. As our Subtask2b architecture relied more on the visual modality than Subtask2a, the reduction of performance is therefore expected given this analysis.

7.3 Post-Evaluation Analysis

Post official evaluation, we used our analysis of the competition results to explore an improved architecture for each task. Whilst these are *not* part of the official SemEval Task 4 leaderboard, we include these as additional experiments.

For Subtask2a, we incorporated the VGG19

model instead of CLIP and removed the second mBERT model with the 80% drop-out rate with the aim to provide more information from the visual modality. For Subtask2b, we attempted to improve the linguistic part of the model by incorporating XLM-Roberta-Large.

Additionally, for Subtask2b, we direct translated 200 memes per test language from the Memotion (Sharma et al., 2020) dataset which were considered 'not offensive' and labelled these non-propagandistic, to significantly increase and re-balance the data provided for Subtask2b. In this new augmented data, each test language comprised 10% of the non-propagandistic label whereas English comprised 70%, also drawing memes from Memotion in English to balance the label sample size.

Despite incorporating the visual modality and additional data, our second attempt at Subtask2a under-performed. Considering the drop, we did not feel the inclusion of external knowledge via an additional BERT model as in prior experiments would improve performance. Since our augmentation technique cannot replicate the visual modality, the visual information contains cultural entities and concepts from English-memes which likely impacts performance, particularly for techniques that require more contribution from the visual modality.

In Subtask2b, all languages improved without BERT(ex). Performance on Arabic decreased slightly with the inclusion of external knowledge, with no change in Bulgarian and an increase in North Macedonian. The inclusion of external knowledge via late fusion, comparative to the results in Table 4, provided marginal improvement; likely the dataset re-balance and inclusion of a larger language model were also significant. The augmented data for this experiment were also more diverse in this case as they were drawn from a different dataset, whereas augmenting the multilabel classes in Subtask2a from another dataset was not possible without native language speakers trained in the specific annotation task.

Subtask2a	Test Language	F1	F1 Change
mBERT+XLM-RBase + VGG19	Bulgarian	0.424	-0.059
mBERT+XLM-RBase + VGG19	North Macedonian	0.358	-0.156
mBERT+XLM-RBase + VGG19	Arabic	0.376	-0.040
Subtask2b			
XLM-RL + VGG19	Bulgarian	0.571	0.065
XLM-RL + VGG19	North Macedonian	0.570	0.135
XLM-RL + VGG19	Arabic	**0.621**	0.111
XLM-RL + VGG19 + BERT(ex)	Bulgarian	0.571	0.065
XLM-RL + VGG19 + BERT(ex)	North Macedonian	0.578	**0.143**
XLM-RL + VGG19 + BERT(ex)	Arabic	0.603	0.093

Table 8: Post-Evaluation Model Results

8 Conclusion and Future Work

We presented our ensemble learning approach to SemEval-2024 Task 4, including a number of experiments with early and late fusion, the inclusion of external knowledge and modifying the label threshold. We found that the inclusion of external sources of knowledge, even basic descriptive entities as in our experiments, improved performance on both subtasks especially using late fusion.

By their nature, memes are multimodal; our approach to Subtask2a still utilised visual elements via entities extracted from the image, and thus provided essential context to interpret ambiguous textual content, however we found the balance between visual and textual importance varied across meme types and tasks. Whilst Subtask2a benefited from the integration of visual entities as a more concise representation of the visual modality, we found that much of the context required for identifying specific techniques required either better cross-modal understanding or finer text analysis. In contrast, Subtask2b benefited from a strong visual model.

The identification of named entities in visual modality of memes is a potential future area of research, as this would enable drawing on complex stores of knowledge (e.g., knowledge graphs) for deeper cross-modal understanding when disentangling persuasion techniques. We further suggest that there is promise in generating more high quality, multilingual data for persuasion techniques across languages based on our experiments with augmented data, particularly for low-resource languages. Although we augmented the task data to cover more languages using direct translation, a limitation in this method is the inability to change the visual modality.

We also note there is a cultural element to memes not considered in current research. We identified that North Macedonian memes were visually different from other memes; the different cultural perspectives and practices in developing memes is under-researched, with only limited studies investigating global meme practices (Nissenbaum and Shifman, 2018). As well varied training data, a better understanding of cultural meme production could contribute to defining the most appropriate approach for zero-shot multilingual meme tasks.

Acknowledgements

We acknowledge the Viper High Performance Computing facility of the University of Hull and its support team.

References

Tariq Habib Afridi, Aftab Alam, Muhammad Numan Khan, Jawad Khan, and Young-Koo Lee. 2021. A multimodal memes classification: A survey and open research issues. In *Innovations in Smart Cities Applications Volume 4: The Proceedings of the 5th International Conference on Smart City Applications*, pages 1451–1466. Springer.

David M Beskow, Sumeet Kumar, and Kathleen M Carley. 2020. The evolution of political memes: Detecting and characterizing internet memes with multi-modal deep learning. *Information Processing & Management*, 57(2):102170.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Lydia Bryan-Smith, Jake Godsall, Franky George, Kelly Egode, Nina Dethlefs, and Dan Parsons. 2023. Real-time social media sentiment analysis for rapid impact assessment of floods. *Computers Geosciences*, 178:105405.

Rui Cao, Ming Shan Hee, Adriel Kuek, Wen-Haw Chong, Roy Ka-Wei Lee, and Jing Jiang. 2023. Procap: Leveraging a frozen vision-language model for hateful meme detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5244–5252.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021a. Detecting propaganda techniques in memes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6603–6617, Online. Association for Computational Linguistics.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021b. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Marc J Dupuis and Andrew Williams. 2019. The spread of disinformation on the web: An examination of memes on social networking. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1412–1418. IEEE.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Eftekhar Hossain, Omar Sharif, and Mohammed Moshiul Hoque. 2022. MemoSen: A multimodal dataset for sentiment analysis of memes. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1542–1554, Marseille, France. European Language Resources Association.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Asaf Nissenbaum and Limor Shifman. 2018. Meme templates as expressive repertoires in a globalizing

world: A cross-linguistic study. *Journal of Computer-Mediated Communication*, 23(5):294–310.

Jakub Piskorski, Nicolas Stefanovitch, Valerie-Anne Bausier, Nicolo Faggiani, Jens Linge, Sopho Kharazi, Nikolaos Nikolaidis, Giulia Teodori, Bertrand De Longueville, Brian Doherty, Jason Gonin, Camelia Ignat, Bonka Kotseva, Eleonora Mantica, Lorena Marcaletti, Enrico Rossi, Alessio Spadaro, Marco Verile, Giovanni Da San Martino, Firoj Alam, , and Preslav Nakov. 2023a. News categorization, framing and persuasion techniques: Annotation guidelines. technical report jrc-132862. European Commission Joint Research Centre.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023b. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

Chhavi Sharma, Deepesh Bhageria, William Paka, Scott, Srinivas P Y K L, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 Task 8: Memotion Analysis-The Visuo-Lingual Metaphor! In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain. Association for Computational Linguistics.

Shivam Sharma, Firoj Alam, Md. Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty. 2022. Detecting and understanding harmful memes: A survey. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5597–5606. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Victoria Sherratt. 2022. Towards contextually sensitive analysis of memes: Meme genealogy and knowledge base. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5871–5872. International Joint Conferences on Artificial Intelligence Organization. Doctoral Consortium.

Victoria Sherratt, Kevin Pimbblet, and Nina Dethlefs. 2023. Multi-channel convolutional neural network for precise meme classification. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*, pages 190–198.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Amit Singhal. 2012. Introducing the knowledge graph: Things, not strings.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

Riccardo Tommasini, Filip Ilievski, and Thilini Wijesiriwardene. 2023. Imkg: The internet meme knowledge graph. In *The Semantic Web: 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28–June 1, 2023, Proceedings*, page 354–371, Berlin, Heidelberg. Springer-Verlag.

Ron Zhu. 2020. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. *CoRR*, abs/2012.08290.

nowhash at SemEval-2024 Task 4: Exploiting Fusion of Transformers for Detecting Persuasion Techniques in Multilingual Memes

Abu Nowhash Chowdhury[1] **and Michal Ptaszynski**[2]
[1]Asian University for Women, Chattogram 4000, Bangladesh
[2]Kitami Institute of Technology, Kitami 090-8507, Japan
nowhash.chowdhury@auw.edu.bd and michal@mail.kitami-it.ac.jp

Abstract

Nowadays, memes are considered one of the most prominent forms of medium to disseminate information on social media. Memes are typically constructed in multilingual settings using visuals with texts. Sometimes people use memes to influence mass audiences through rhetorical and psychological techniques, such as causal oversimplification, name-calling, and smear. It is a challenging task to identify those techniques considering memes' multimodal characteristics. To address these challenges, SemEval-2024 Task 4 introduced a shared task focusing on detecting persuasion techniques in multilingual memes. This paper presents our participation in subtasks 1 and 2(b). We use a finetuned language-agnostic BERT sentence embedding (LaBSE) model to extract effective contextual features from meme text to address the challenge of identifying persuasion techniques in subtask 1. For subtask 2(b), We finetune the vision transformer and XLM-RoBERTa to extract effective contextual information from meme image and text data. Finally, we unify those features and employ a single feed-forward linear layer on top to obtain the prediction label. Experimental results on the SemEval 2024 Task 4 benchmark dataset manifested the potency of our proposed methods for subtasks 1 and 2(b).

1 Introduction

Modern social media represents a prominent environment to disseminate information to a vast community in real time. Hence, persuasion techniques are often embedded in social media content to subliminally influence people and their unconscious opinions. Such techniques are now incorporated in memes due to the increasing popularity among social media users. The visual aspect of memes adds to the effectiveness of grabbing people's attention than purely word-based messages. Manipulators and propagandists now treat it as an effective tool to promote and achieve their nefarious agendas. Sometimes different organizations use it to spread fake news or propaganda which causes social chaos and incitement of hate, which could result in harm or even human casualties. Hence, the detection of persuasion techniques embedded in memes appears as a formidable task to shield individuals from deceit. Moreover, detecting these techniques from memes is a challenging task since it requires a nuanced understanding of images, and texts, and a proper appreciation of the satirical characteristics of memes. To address these challenges, SemEval-2024 introduced a shared task focusing on detecting persuasion techniques from multilingual memes (Dimitrov et al., 2024). This task comprises three subtasks. Whereas the first task is based on identifying 20 persuasion techniques from meme texts. This is a hierarchical multilabel text classification task. Tasks 2(a) and 2(b) are based on multi-modal contents. Task 2(a) is a hierarchical multimodal multilabel classification task where the proposed system needs to identify 22 persuasion techniques from multimodal memes. Task 2(b) is a multimodal binary classification task where the participants need to apply the multimodal information expressed by memes to classify them into whether they contain a persuasion technique or not. A data sample of each task along with corresponding labels was articulated in Table 1.

However, some prior works have been done on identifying persuasion techniques from texts and visuals. SemEval 2023 shared task 3 introduced a subtask based on identifying persuasion techniques used in news articles (Piskorski et al., 2023). Most of the participants used different multilingual transformer models to tackle the challenge of this task. APatt (Purificato and Navigli, 2023) utilized an ensemble of different pre-trained transformer models e.g., XLNet, RoBERTa, BERT, ALBERT, and De-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 133–138

Table 1: Sample Data of subtask 1, 2(a), and 2(b) of SemEval 2024 Task 4

Task No.	Sample Data	Label
Subtask 1	WHEN THE POWER OF LOVE IS GREATER THAN THE LOVE OF POWER, THE WORLD WILL KNOW PEACE	Loaded Language, Black-and-white Fallacy/Dictatorship, Slogans
Subtask 2(a)	Time To Straighten Out What Is Happening In Our Country! `prop_meme_4398.png`	Flag-waving,Glittering generalities (Virtue),Black-and-white Fallacy/Dictatorship
Subtask 2(b)	I MISSED THE SUPERBOWL-WHO WON?\\nEVERYONE WHO DIDN'T WATCH IT `prop_meme_4388.png`	non_propagandistic

BERTa incorporated by weighted average. Another team, KInITVeraAI (Hromadka et al., 2023) used fine-tuned XLM-RoBERTa-large model to address the multilingual characteristics of this task. They experimented with different prediction threshold values to find the optimal one.

To detect persuasion techniques in texts and images, SemEval 2021 Task 6 introduced three subtasks including multilabel text classification, span identification, and multi-modal multilabel classification task (Dimitrov et al., 2021). The top-performing team on the multilabel text classification task (Tian et al., 2021) leveraged five fine-tuned transformer models: BERT, RoBERTa, XLNet, DeBERTa, and ALBERT. They made use of external PTC corpus (Da San Martino et al., 2020) along with given training data to train these transformer models. Team NLPIITR (Gupta and Sharma, 2021) made use of a fine-tuned RoBERTa model to address the challenge of this task. Team Volta (Gupta et al., 2021) explored the potency of fine-tuned BERT and RoBERTa models for both multi-label text classification and span identification tasks and used RoBERTa Large for the final model. Their proposed architecture ranked top on span identification tasks. For the multi-label multimodal classification task, they tested the performance of the ensemble of multimodal transformers e.g., UNITER, VisualBERT, and LXMERT alongside unimodal transformers e.g., BERT, and RoBERTa. The winning team on subtask 3 (Feng et al., 2021) experimented with the ensemble of fine-tuned DeBERTa and ResNET, DeBERTa and BUTD, and ERNIE-ViL models to address the challenge of leveraging features from different data modalities.

In this paper, we demonstrate our proposed architecture to address the challenges of Subtask 1 (multi-label hierarchical text classification) and Subtask 2(b) (multi-modal binary classification) of SemEval 2024 Task 4. For subtask 1, we utilize a fine-tuned Language-agnostic BERT Sentence Embedding (LaBSE) model to extract effective contextual features of meme texts. Next, we utilize an ensemble of Vision Transformer and XLM-RoBERTa models to address the challenge of multilingual and multi-modal characteristics of subtask 2(b).

The remaining part of the manuscript is outlined as follows: The pictorial description of our proposed methods for both tasks is articulated in Section 2. Section 3 presents the experimental setup, result, and evaluation. We conclude this manuscript with some future research directions in Section 4.

2 Proposed Architecture

2.1 Subtask 1: Hierarchical Multi-label Persuasion Techniques Classification from Meme Text

The main objective of our system is to detect available persuasion techniques in meme text from 20 pre-defined persuasion technique categories. An overview of our proposed persuasion technique detection framework is shown in Figure 1.

Upon obtaining the meme texts, we employed Language-agnostic BERT sentence embedding (LaBSE) on top of Flair's Transformer Document Embeddings to generate effective document embedding vectors. Further, those document vectors are then fed to a single-layer feed-forward linear classifier to obtain the prediction label.

2.1.1 Language-agnostic BERT Sentence Embedding (LaBSE)

LaBSE is a multilingual transformer-based Language-agnostic BERT Sentence Embedding model developed by (Feng et al., 2020). It was trained on 6 Billion translation pairs and can generate sentence-level shared embedding features for 109 languages. To obtain optimal representations of multilingual sentences, LaBSE integrates

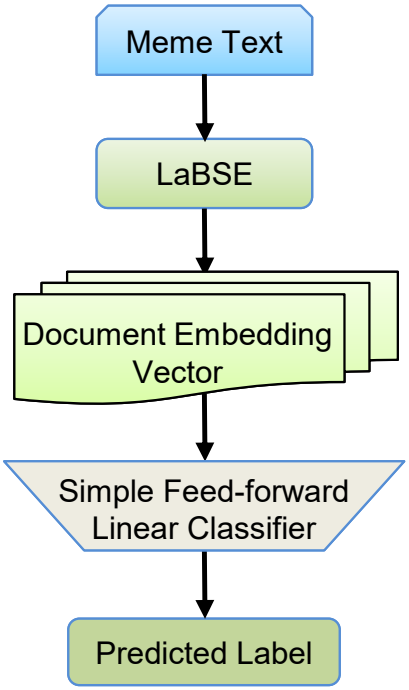


Figure 1: Proposed Framework of Subtask 1.

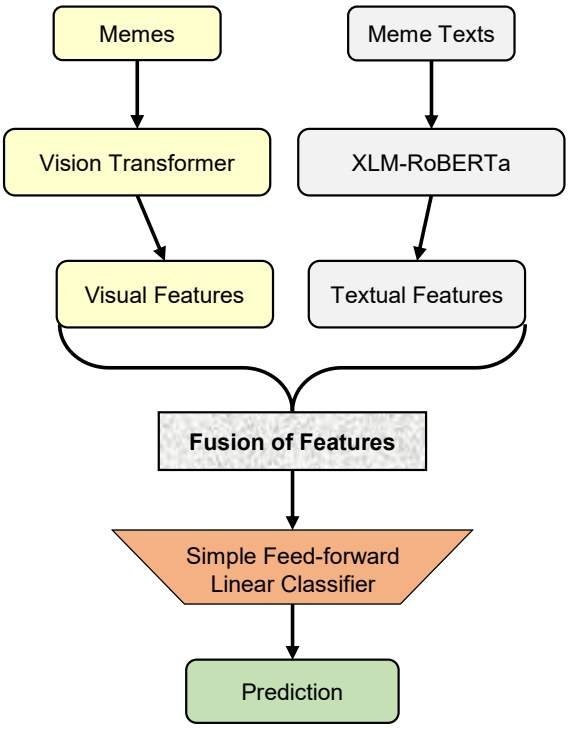


Figure 2: Proposed Framework of Subtask 2(b).

both monolingual and cross-lingual representations. It incorporates Multilingual BERT utilizing the masked language model and transformer language model with a translation ranking task alongside bidirectional dual encoders. We finetuned the LaBSE model on the benchmark dataset to capture the task-specific context effectively.

2.1.2 Transformer Document Embeddings

Document embedding represents embedding features of a full sentence rather than individual tokenized features. Flair's transformer document embeddings (Akbik et al., 2019) furnish an embedding for the entire text. We can extract embeddings directly from a pre-trained transformer model for a full sentence which enables us to capture the context of a sentence effectively. In our proposed architecture, we leverage the LaBSE model with transformer document embedding to obtain sentence-level embedding for a particular meme text.

2.2 Subtask 2: Multimodal Binary Classification Task

Figure 2 illustrates our proposed framework for subtask 2(b) where we tackled the challenges of multimodal meme classification.

Upon obtaining meme images and meme texts, we utilize a vision transformer and XLM-RoBERTa model to extract embedding features for both the meme images and meme texts. To tackle the multimodal characteristics of this task, we then integrated both visual and textual embedding features together and fed the integrated features to a single-layer feed-forward linear classifier to obtain the final prediction label.

2.2.1 Vision Transformer

The vision transformer (ViT) is a self-supervised transformer encoder model pre-trained on a large image corpus (Dosovitskiy et al., 2020). ViT generates lower dimensional linear embedding by splitting the input image into fixed-size patches and flattening the patches. After adding positional embeddings, the flattened patches are fed into a standard transformer encoder as a sequence of tokens. The ViT encoder's internal architecture is similar to that of the original transformer. We utilize the finetuned ViT model facebook/dino-vitb16 checkpoint[1] (Caron et al., 2021) to extract effective visual information from memes.

2.2.2 XLM-RoBERTa

XLM-RoBERTa is a cross-lingual sentence encoder introduced by the Facebook AI group (Conneau et al., 2019). It was trained on a large 2.5 TB Common Crawl(CC) corpus containing over 100

[1]https://huggingface.co/facebook/dino-vitb16

languages. XLM-RoBERTa showed SOTA performance in various cross-lingual tasks (Eronen et al., 2022, 2023b,a). Both the base and large variants of XLM-RoBERTa contain 250M and 560M parameters, respectively with 250K vocabulary. In our proposed multimodal architecture, we utilized the finetuned XLM-RoBERTa large version to extract an effective representation of meme texts.

2.2.3 Fusion of Features

The fusion of high-level features from different data modalities in a neural architecture is conventional to tackle the challenge of representing multimodal features (Kumar and Nandakumar, 2022), (Pramanick et al., 2021), (Velioglu and Rose, 2020). In our proposed multimodal framework, we concatenate visual and textual features extracted from the finetuned ViT and the finetuned XLM-RoBERTa model for the effective representation of multimodal features.

2.3 Prediction Module

For both subtasks 1 and 2(b), We employed a single-layer feed-forward linear layer with SoftMax activation function to obtain the prediction, like in the equation 1 below.

$$q = Wp + b \tag{1}$$

Here, the input and output feature vectors are represented by p and q respectively. W is the weight matrix and b indicates the bias.

3 Experiments

3.1 Dataset Description

For subtasks 1 and 2(b), we utilized the dataset provided by the SemEval 2024 Task 4 organizers (Dimitrov et al., 2024) to train and finetune our proposed frameworks. Table 2 shows the detailed statistics of the dataset.

To evaluate the performance of our proposed frameworks, we utilized the hierarchical F1 score for subtask 1 and macro F1 score for subtask 2(b) as per the benchmark of SemEval 2024 Task 4 (Dimitrov et al., 2024).

3.2 Experimental Setup

We utilized the Google Colaboratory platform for system implementation, training, parameter tuning, and performance analysis. For subtask 1, we utilized the LaBSE model on Flair's NLP framework. The parameters used to train and finetune our model are illustrated in Table 3.

We made use of the vision transformer and XLM-RoBERTa model to tackle the challenge of subtask 2(b). The parameters used to train the vision transformer and XLM-RoBERTa are shown in Table 4 and Table 5, respectively.

Table 2: The statistics of the dataset.

Language	#Train	#Val	#Dev	#Test
Subtask 1:				
English	7000	500	1000	1500
Bulgarian	-	-	-	426
North Macedonian	-	-	-	259
Arabic	-	-	-	100
Subtask 2(b):				
English	1200	150	300	600
Bulgarian	-	-	-	100
North Macedonian	-	-	-	100
Arabic	-	-	-	160

Table 3: Optimal parameter settings for subtask 1.

Parameters List	Search Space	Value
Epochs	{4}	4
Batch size	{4}	4
Learning rate	{5e-5}	5e-5
Optimizer	{Adam, MADGRAD}	Adam
Multi-label Threshold	{0.1, 0.2, 0.30}	0.1

Table 4: Optimal parameter settings used in Vision Transformer.

Parameters List	Search Space	Value
Epochs	{4,6,8}	8
train_batch_size	{4,8,16}	8
eval_batch_size	{4,8,16}	8
Learning rate	{4e-5, 5e-5, 6e-5}	6e-5
Optimizer	{AdamW}	AdamW

Table 5: Optimal parameter settings used in XLM-RoBERTa.

Parameters List	Search Space	Value
Epochs	{4,6,8}	6
train_batch_size	{4,8,16}	4
eval_batch_size	{4,8,16}	4
Learning rate	{4e-5, 5e-5, 6e-5}	6e-5
Optimizer	{AdamW}	AdamW

Table 6: Synopsis of our proposed system performance in subtask 1.

Language	Hierarchical		
	F1 score	Precision	Recall
English	0.64096	0.61167	0.67320
Bulgarian	0.48627	0.46007	0.51563
North Macedonian	0.42558	0.41395	0.43788
Arabic	0.40370	0.35989	0.45965

Table 7: Synopsis of our proposed system performance in subtask 2(b)

Language	F1 Macro	F1 Micro
English	0.49845	0.51500
Bulgarian	0.43363	0.45000
North Macedonian	0.42857	0.52000
Arabic	0.49831	0.53125

3.3 Results and Analysis

In this section, we assess the performance of our submitted systems in SemEval 2024 Task 4 subtasks 1 and 2(b). The test dataset comprises four languages including English, Bulgarian, North Macedonian, and Arabic. Table 6 and Table 7 illustrate the performance of our model for subtasks 1 and 2(b), respectively.

For subtask 1, the experimental result shows that our proposed method achieved a good Hierarchical F1 score across the English, Bulgarian, North Macedonian, and Arabic datasets. We also report the hierarchical recall and hierarchical precision scores. This signifies the versatility of our approach across multiple languages. In subtask 2(b), there is still a significant performance gap between the top-performing systems and our system. One plausible reason might be the imbalanced fusion of visual and textual features.

4 Conclusion and Future Works

In this manuscript, we presented our proposed frameworks to address the challenge of SemEval 2024 Task 4 subtasks 1 and 2(b). We employed the LaBSE model to address the multilingual characteristics of subtask 1 whereas Vision Transformer and XLM-RoBERTa models were employed to address the multi-modal and multilingual characteristics of subtask 2(b). Both of our methods showed competitive performance over other participant's systems.

In the future, we aspire to explore the effectiveness of different multimodal transformer models' performance on this task. We also have a plan to exploit the external knowledge for a better understanding of memes for this task.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers

for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Juuso Eronen, Michal Ptaszynski, and Fumito Masui. 2023a. Enhancing cross-lingual learning: Optimal transfer language selection with linguistic similarity. *Science Talks*, 6.

Juuso Eronen, Michal Ptaszynski, and Fumito Masui. 2023b. Zero-shot cross-lingual transfer language selection using linguistic similarity. *Information Processing & Management*, 60(3):103250.

Juuso Eronen, Michal Ptaszynski, Fumito Masui, Masaki Arata, Gniewosz Leliwa, and Michal Wroczynski. 2022. Transfer language selection for zero-shot cross-lingual abusive language detection. *Information Processing & Management*, 59(4):102981.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Zhida Feng, Jiji Tang, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, and Li Chen. 2021. Alpha at semeval-2021 task 6: Transformer based propaganda classification. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 99–104.

Kshitij Gupta, Devansh Gautam, and Radhika Mamidi. 2021. Volta at SemEval-2021 task 6: Towards detecting persuasive texts and images using textual and multimodal ensemble. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1075–1081, Online. Association for Computational Linguistics.

Vansh Gupta and Raksha Sharma. 2021. NLPIITR at SemEval-2021 task 6: RoBERTa model with data augmentation for persuasion techniques detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1061–1067, Online. Association for Computational Linguistics.

Timo Hromadka, Timotej Smolen, Tomas Remis, Branislav Pecher, and Ivan Srba. 2023. KInITVeraAI at SemEval-2023 task 3: Simple yet powerful multilingual fine-tuning for persuasion techniques detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 629–637, Toronto, Canada. Association for Computational Linguistics.

Gokul Karthik Kumar and Karthik Nandakumar. 2022. Hate-clipper: Multimodal hateful meme classification based on cross-modal interaction of clip features. *arXiv preprint arXiv:2210.05916*.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. Momenta: A multimodal framework for detecting harmful memes and their targets. *arXiv preprint arXiv:2109.05184*.

Antonio Purificato and Roberto Navigli. 2023. APatt at SemEval-2023 task 3: The sapienza NLP system for ensemble-based multilingual propaganda detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 382–388, Toronto, Canada. Association for Computational Linguistics.

Junfeng Tian, Min Gui, Chenliang Li, Ming Yan, and Wenming Xiao. 2021. Mind at semeval-2021 task 6: Propaganda detection using transfer learning and multimodal fusion. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1082–1087.

Riza Velioglu and Jewgeni Rose. 2020. Detecting hate speech in memes using multimodal deep learning approaches: Prize-winning solution to hateful memes challenge. *arXiv preprint arXiv:2012.12975*.

HalluSafe at SemEval-2024 Task 6: An NLI-based Approach to Make LLMs Safer by Better Detecting Hallucinations and Overgeneration Mistakes

Zahra Rahimi, Hamidreza Amirzadeh, Alireza Sohrabi,
Zeinab Sadat Taghavi and Hossein Sameti
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
{zarahimi, hamid.amirzadeh78, sameti}@sharif.edu
{lirezasohrabi, zeinabtaghavi1377}@gmail.com

Abstract

The advancement of large language models (LLMs), their ability to produce eloquent and fluent content, and their vast knowledge have resulted in their usage in various tasks and applications. Despite generating fluent content, this content can contain fabricated or false information. This problem is known as hallucination and has reduced the confidence in the output of LLMs. In this work, we have used Natural Language Inference to train classifiers for hallucination detection to tackle SemEval-2024 Task 6-SHROOM (Mickus et al., 2024) which is defined in three sub-tasks: Paraphrase Generation, Machine Translation, and Definition Modeling. We have also conducted experiments on LLMs to evaluate their ability to detect hallucinated outputs. We have achieved 75.93% and 78.33% accuracy for the model-aware and model-agnostic tracks, respectively. The shared links of our models and the codes are available on GitHub[1].

1 Introduction

Large language models are compelling in content generation. The ability of these models has led to their widespread use in various applications. Some of the use cases of these models are in sensitive fields, such as consulting in medicine and law. The eloquence of LLMs makes their content appear very acceptable, and these models respond with high confidence. An important shortcoming of these models is hallucination. Hallucination is the production of fabricated or false content (Gehman et al., 2020; Weidinger et al., 2021). Hallucination detection and mitigation are necessary to avoid the dangers of spreading false and harmful information. According to Zhang et al. (2023), hallucinations can be divided into input hallucinations, context hallucinations, and factual hallucinations. In input hallucination, the output content of the model has data that contradicts the input content. In context hallucination, the model's output content contradicts the content the model itself produced earlier. In the last case, factual hallucination, the output content of the model has information that contradicts the existing world knowledge. In the dataset provided for the Shroom task, each data sample has a reference to be checked with. Given that reference-based hallucination detection entails identifying contradictions between model output and the reference (either input or target), a natural language inference (NLI) approach presents an intuitive solution to detect such contradictions and consequently identify instances of hallucination, therefore we adopt an NLI approach as the foundation of our methodology.

Through this task, we have gained knowledge about hallucinations, their causes, and the various approaches to deal with them. Language model responses can be so fluent that it becomes difficult even for a human agent to detect hallucinations. Therefore, it is essential to train these models to recognize the limits of their knowledge. If they lack sufficient understanding of a subject, they should search for reliable sources and inform the human user if they are unsure of their answer. Our team ranked 19th and 30th in the model-aware and model-agnostic tracks, respectively, with a difference of 2.93% and 8.4% compared to the top-ranked team. We found that the decision boundary for detecting hallucinations can be very narrow in some cases. While our system has shown relatively good performance, there is still room for improvement.

2 Background

As mentioned earlier, there are three types of hallucinations. The types of hallucinations considered in this task are "factual" and "input". The "factual"

[1] https://github.com/z-rahimi-r/HalluSafe-at-SemEval-Task-6-SHROOM

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 139–147

type occurs in the definition modeling task, where the definition of a word or phrase must be provided, and the "input" type appears in the paraphrase generation and machine translation tasks. The hallucination detection track has two sub-tracks: model-aware and model-agnostic. In the model-aware subtrack, the model that generated the data is specified, and participants can use model parameters for diagnosis or analysis. However, our approach assumes the models are black-box and can be used for situations where we do not have access to the internal states and parameters of the model. It is important to note that overgeneration is another issue in LLM outputs. Samples with this issue should also be labeled as One, indicating the presence of hallucinations. Hallucination is not specific to LLMs, and before the emergence of these models, it has been investigated in NLP tasks such as summarization and machine translation (Azaria and Mitchell, 2023).

To deal with the hallucination problem in LLMs, it is essential to find the causes of the problem first. Two probable causes of hallucination, stated in Azaria and Mitchell (2023), are the model focusing on producing one token each time and random sampling to increase diversity in text production. Some believe overfitting to training data may lead to hallucination (McKenna et al., 2023). In contrast to this point of view, in Yao et al. (2023), they have shown that prompts consisting of only random meaningless tokens can also elicit hallucinations in LLMs. They believe that hallucinations are beyond training data and consider them as adversarial features. They have observed in their experiments that a slight change in the original prompt can produce a completely different claim by the LLM, which indicates that LLMs are very non-robust. In Rawte et al. (2023), they measure the relationship between linguistic factors such as readability, formality, and concreteness of prompts and hallucinations. Their results show that more concrete and formal prompts lead to fewer hallucinations, but no definite conclusion can be drawn regarding the effect of readability on hallucinations. According to this article, prompt engineering can be effective in reducing the problem of hallucinations. Lengthy prompts can hurt the understanding of the LLM. In some experiments, it has been observed that the LLM performs better when the critical information is placed at the beginning or end of the prompt. The performance quality decreases when the model needs to access the middle parts of the prompt for information.

Hallucination can be mitigated in different stages of an LLM's life cycle. As we know, the life cycle of an LLM consists of Pre-training, SFT (Supervised Fine-Tuning), RLHF (Reinforcement Learning with Human Feedback), and Inference (Zhang et al., 2023). The datasets with which LLMs are pre-trained are collected without human supervision. These data can include false or outdated information, which may cause hallucinations. The training in the SFT phase should also consider the knowledge of the model, and the model should not be fine-tuned for an application that has not acquired sufficient knowledge during the pre-training. One way to reduce hallucinations in both the SFT and RLHF phases is to teach the model to be honest. The language model should be trained to avoid commenting on a subject if it does not have enough information (Zhang et al., 2023). The methods investigated in this work are related to detecting and mitigating hallucination in the inference phase. The related previous works can be categorized as white box, gray box, and black box depending on the level of access to internal parameters of the LLM. The methods that use the internal state of the language model for diagnosis are white-box approaches. Gray box approaches are methods that access the output distribution of the model, such as detecting hallucinations at the token level. Finally, Blackbox approaches only have access to the textual output of the model.

2.1 White-Box Approaches

In Azaria and Mitchell (2023), the SAPLMA approach (Statement Accuracy Prediction, based on Language Model Activations) has been introduced. Their approach uses the internal state of the LLM to measure the truthfulness of the statements. This applies to both the statements provided to the model and the statements produced by the model itself. They use a relatively shallow feedforward network as a classifier, which measures the truthfulness probability of a statement based on the values of the hidden layer activators.

2.2 Gray-Box Approaches

These approaches use the uncertainty of models to detect hallucinations. The idea of these approaches is that when the model is sure of the correctness of a sentence, the distribution probability of tokens of the sequence is sharp. Still, in uncertain conditions, this distribution will probably be flat. Kadavath et al. (2022) suggests that a model's confidence in

answering a specific question correlates with the certainty of its response. They propose repeatedly sampling the answer at T = 1, yielding an answer distribution characterized by low entropy when the model is confident. Conversely, when the model is uncertain, it tends to produce "hallucinated" responses, resulting in an answer distribution with high entropy. Nevertheless, experimental results indicate that utilizing entropy as a metric for determining whether a model knows the answer to a question is not consistently reliable, particularly as models scale in size. Another work in this group of methods is Yuan et al. (2021), in which a score named BART-Score evaluates the text's quality generated by the model from different aspects such as informativeness, fluency, and factuality. Using token-level probabilities, BART-Score calculates the probability of an output sequence given a specific input sequence.

2.3 Black-Box Approaches

The methods presented in Martino et al. (2023) and Manakul et al. (2023) are black box methods. In Martino et al. (2023), where a large language model is used for the" Review Response" task, the knowledge injection method adds related information to the prompt. The relevant knowledge is extracted from a knowledge graph specific to that particular business. It includes information such as addresses, phone numbers, etc., which are naturally not available in the training data of an LLM. The target hallucination in this task is factual. Fact-based verification methods require an external database, and their inference is computationally expensive. The introduced method in Manakul et al. (2023) uses no external knowledge source. Their approach, self-checkGPT, is based on the idea that if an LLM knows a subject, sampled responses do not contradict each other. The proposed approach has five variants: BERTScore, question-answering, n-gram, NLI, and LLM prompting. The best-performing variant is LLM prompting, in which they ask an LLM if a sentence is supported by a context or not. This variant has a high computational cost. The second best is the NLI variant, which uses natural language inference to detect inconsistency between sampled responses.

In Mündler et al. (2023), a prompting-based framework is introduced to efficiently identify and address instances of self-contradiction, meaning context hallucinations. Their investigation delved into open-domain text generation utilizing a dual-LM setup: one LM for text generation and another as an analyzer. For each sentence generated by the initial LM, a corresponding sentence is produced based on the associated context, and both are subsequently subjected to analysis by the second LM. In cases where the analyzer LM identifies a contradiction between the two sentences, it is prompted to revise the given sentences and remove the contradiction so that the output is informative and coherent with the corresponding context. ChainPoll (Friel and Sanyal, 2023) represents another recent advancement in addressing hallucinatory phenomena within LLMs. The approach adopted for hallucination detection is straightforward: employing a carefully crafted prompt, the authors prompt the GPT-3.5-turbo model to assess whether the completion contains hallucinations driven by a chain of thought (CoT) explanation. Iterating this process several times and aggregating the "yes" responses yields a probability score ranging from Zero to One, indicating the likelihood of hallucination.

In Guerreiro et al. (2023), hallucinations in translation models are studied concerning two different sources: perturbations and natural hallucinations. Hallucinations induced by perturbations occur when the model memorizes the training data and outputs a faulty translation triggered by a slight change in the input sequence. In contrast, natural hallucinations occur due to poor quality of training data. Natural hallucinations are divided into two categories (Raunak et al., 2021): detached and oscillatory. In the detached type, the output is fluent but inadequate. In the oscillatory type, the output has repeated n-grams. In this article, a black box method (Top N-Gram (Raunak et al., 2021)) and a white box method (ALTI+ (Ferrando et al., 2022)) have been used to detect natural hallucinations. It has been observed that hallucinations in translations occur more often for low-resource languages. Another work concerning detecting machine translation hallucinations is COMET (Rei et al., 2020), a reference-based neural framework with superior performance compared to conventional approaches (Guerreiro et al., 2022). It has two architectures, one of which is an estimator model, which tries to directly regress on human judgment scores for quality assessment. In contrast, the other one, a ranking model, minimizes the distance between a "better" hypothesis and its corresponding reference and original source translations.

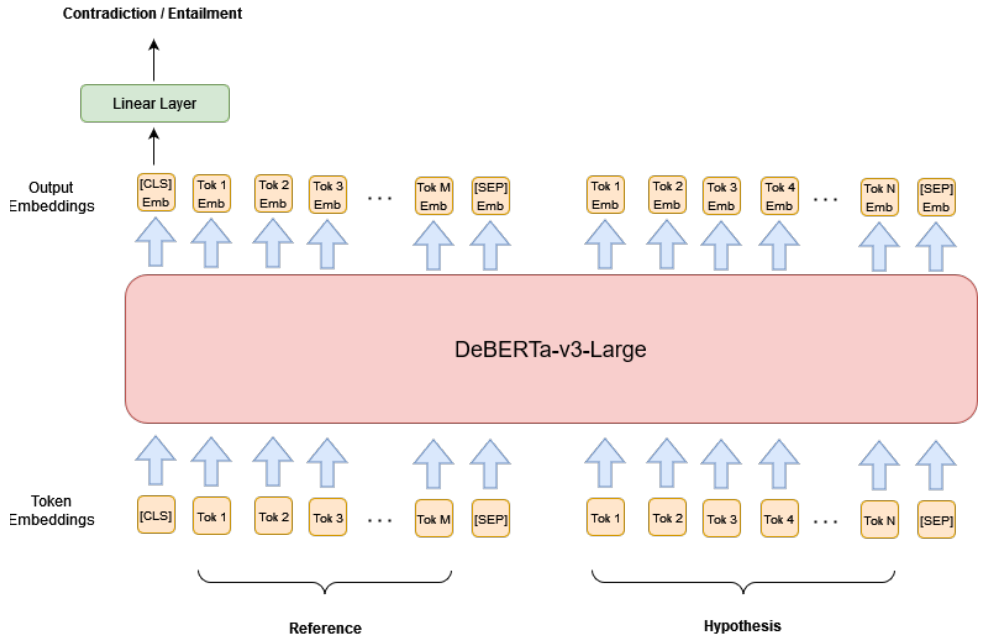


Figure 1: Formulating hallucination detection problem as an NLI task

	DM	PG	MT	Total
Train	20000	20000	20000	60000
Dev	375	250	375	1000
Trial	36	9	35	80
Test	1125	750	1125	3000

Table 1: Dataset Statistics

3 System Overview

In this section, we introduce our proposed system. The general system sketch is presented in Figure 1. Additionally, detailed statistics regarding the dataset are outlined in Table 1. Since the training data provided for this task was unlabeled, we labeled 3000 samples of the training data. Since LLMs have hallucination problems themselves, the labeling was done by a human agent. We have trained separate models for each task (MT, PG, and DM) to detect hallucinations. The model is DeBERTa-v3 large (He et al., 2023) and was first trained on the NLI task and then fine-tuned on the labeled data of each task. Finally, the model with the highest accuracy on validation data was saved. For training a binary classification model on the NLI task, only the data samples with labels of contradiction and entailment of the NLI dataset of Stanford University (Bowman et al., 2015) were used.

Examples of data samples for PG, MT, and DM tasks are presented in Table 2. Each sample has a source, target, and hypothesis in the MT task. The source sentence may be in languages other than English, but the target sentence is always in English. In the PG task, each sample has a source and hypothesis. We can detect hallucinations in these two tasks using the target sentence as the reference for the MT task and the source sentence as the reference for the PG task. Since the nature of hallucination in the PG and MT tasks is almost the same, the training data of both tasks were used to train the model for these two tasks. For Each task, the model with the highest accuracy on validation data was saved. The sequence classification method is utilized to detect hallucinations. The reference sentence is placed at the beginning, followed by the hypothesis sentence, separated with a "[SEP]" token. The hypothesis is the output of the LLM that may contain hallucinations. Finally, the entire sequence is fed into the NLI model, which outputs probabilities for each class, contradiction, and entailment. If the hypothesis contains information that contradicts the reference, the output label of our NLI model should be equal to 1, indicating contradiction. The probability of contradiction is considered equivalent to the probability of hallucination.

In addition to training classifier models, we have conducted tests to evaluate the performance of

PG	**src**	The budget cannot be adopted against the will of the European Parliament.
	hyp	The European Parliament does not approve the budget.
	label	Not Hallucination
MT	**src**	Doonii fayyadamuun meeshaa geejibuun namootabaay'ee fi meeshaalee galaanarra cesisuuf karaa baayee si'aataa dha.
	tgt	Using ships to transport goods is by far the most efficient way to move large amounts of people and goods across oceans.
	hyp	Using a gas-fired device is a way to stop people from using natural gas and other equipment.
	label	Hallucination
DM	**src**	Communistic birds. What is the meaning of communistic?
	tgt	Living or having their nests in common.
	hyp	Of or pertaining to communism.
	label	Hallucination

Table 2: Data samples of PG, MT, and DM tasks

two large language models, Falcon-7B and chat-GPT3.5, on the hallucination detection task. For this purpose, we have instruction-fine-tuned the falcon-7B model on the labeled training and validation data. For chat-GPT3.5, the accuracy was calculated on the trial set using zero and two-shot inference. For these two models, only the results on the trial set were presented.

We also thought we might find a meaningful connection between token probabilities in the output sequence and hallucination. For this, we took the top token probabilities of the output sequence of the LLM (PG, MT, and DM LLMs) with their labels. We fed them as input to an RNN model, such as LSTM, to predict hallucination based on model uncertainty of token probabilities. Unfortunately, we found out that when outputting hallucinated output, the model is as confident as non-hallucinated ones, and the classifier model could learn absolutely nothing from the token probabilities, no matter how we change the model complexity or hyperparameters.

4 Experimental Setup

To provide enough labeled data to train our models, a total of 3000 of the model-aware and model-agnostic training samples were labeled. Different data splits were tested to get the best accuracy on each task. The details of the split used to train the model with the best accuracy for each task are specified in colab notebooks on GitHub[1]. The results

[1] https://github.com/z-rahimi-r/HalluSafe-at-SemEval-Task-6-SHROOM

	DM	PG	MT
#Samples	36	9	35
Falcon7B 2-shot	47.22	44.44	45.71
Falcon7B 4-shot	33.33	55.55	48.57
Falcon7B finetuned	41.66	66.66	0.4
ChatGPT3.5 zero-shot	86.11	65.71	44.44
ChatGPT3.5 2-shot	86.11	74.28	88.88
ChatGPT3.5 4-shot	83.33	**82.85**	66.66
Best-DM-DeBERTa	**94.44**	55.55	85.71
Best-PG-DeBERTa	86.11	77.77	77.14
Best-MT-DeBERTa	91.66	55.55	**94.28**

Table 3: Results on Trial set

of the trial set are presented in Table 3.

All three models are trained for ten epochs with a learning rate equal to 2e-5 and batch size equal to eight samples. The base model for all three tasks is DeBERTa-v3-large (He et al., 2023), trained on the NLI task with two classes of contradiction and entailment. We have used the Hugging-Face transformers library (Wolf et al., 2020) to train DeBERTa models implemented with PyTorch. For instruction fine-tuning the Falcon-7B model, we also used the Hugging-Face library and LoRA method (Hu et al., 2022). The prompt used for fine-tuning Falcon and inference from chatGPT is similar to that used in the selfCheckGPT (Manakul et al.,

PG	**src**	A five, six, seven, eight.
	tgt	And 5, 6, 7, 8.
	hyp	A number between five and eight.
	gold label	Hallucination
MT	**src**	Est-ce que tu l'aimes?
	tgt	Do you love him?
	hyp	Do you love her?
	gold label	Not Hallucination
DM	**src**	Haul away, keeping strain on both parts of the halyard so that the <define> pigstick </define> remains vertical as it goes up and doesn't foul the spreaders.
	tgt	(nautical) A staff that carries a flag or pennant above the mast of a sailboat.
	hyp	(nautical) A halyard.
	gold label	Not Hallucination

Table 4: Examples of wrongly classified samples

	acc model-agnostic	rho model-agnostic	acc model-aware	rho model-aware
Baseline	69.66	40.29	74.53	48.78
Nli-only	72.4	59.77	73.93	**56.33**
Best-models	**75.93**	**61.53**	**78.33**	53.74

Table 5: Results on Final Test set

2023). The examples can be found in the Appendix. All notebooks, labeled data, and links to saved models are present on our GitHub.

5 Results

We have achieved 75.93% and 78.33% accuracy for the model-aware and model-agnostic tracks of hallucination detection on final test data. We have ranked 19th and 30th in model-aware and model-agnostic tracks with a 2.93% and 8.4% difference with respect to the first-ranked team in the competition. The accuracies of the best model for each task, along with the accuracy of the base NLI model, are provided in Table 5. Also, examples of wrongly classified samples are provided in Table 4. As you can see the wrongly classified samples are challenging. The problem that exists with some samples of the MT task is that in some cases, relying only on the tgt field may result in a wrong label, and it is necessary also to consider the content of the src field as well. This is true about the MT example presented in the table. In this example, hyp and tgt are both correct translations of the source sentence, but when the content of hyp is evaluated against the tgt, it is wrongly labeled as hallucination.

6 Conclusion

In this work, we have trained classifiers based on Natural Language Inference to detect hallucinated outputs for the two model-aware and model-agnostic subtasks of the SemEval-2024 Task-6-SHROOM (Mickus et al., 2024). We have also conducted experiments to evaluate LLMs' ability to perform this task. The fluency of the output of LLMs makes it difficult even for a human evaluator to recognize the hallucinated output. To train the classifiers, we labeled 3000 training data. Labels may be a little affected by the subjectivity of the annotator, and for future work, it is better to have more than one person label each data sample. Our HalluSafe classifiers have achieved 75.93% and 78.33% accuracy for the model-aware and model-agnostic tracks of hallucination detection on final test data and have outperformed official baselines. Regarding future work, enhancing the quality of training data in the pre-training and fine-tuning stages can effectively reduce hallucinations. Given the potential limitations of storing all necessary information within the memory of models, coupled with the need for regular updates to certain information, it may be beneficial to equip models with

search tools rather than relying solely on memory. It is important to train LLMs during the fine-tuning and instruction-tuning stages to refrain from answering questions if they lack sufficient knowledge on a particular subject, which needs a mechanism to be incorporated into these models to enable them to identify the boundaries of their knowledge.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8756–8769, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *ArXiv*, abs/2310.18344.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

Nuno M. Guerreiro, Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. Hallucinations in Large Multilingual Translation Models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517.

Nuno M. Guerreiro, Elena Voita, and André F. T. Martins. 2022. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. *ArXiv*, abs/2208.05309.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Saurav Kadavath, Tom Conerly, Amanda Askell, T. J. Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Christopher Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (llm) hallucination. In *The Semantic Web: ESWC 2023 Satellite Events*, pages 182–185, Cham. Springer Nature Switzerland.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin T. Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *ArXiv*, abs/2305.15852.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Vipula Rawte, Prachi Priya, S.M. Towhidul Islam Tonmoy, Islam Tonmoy, M Mehedi Zaman, A. Sheth,

and Amitava Das. 2023. Exploring the relationship between llm hallucinations and prompt linguistic nuances: Readability, formality, and concreteness. *ArXiv*, abs/2309.11064.

Ricardo Rei, Craig Alan Stewart, Ana C. Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *ArXiv*, abs/2009.09025.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *ArXiv*, abs/2309.01219.

A Appendix

An example of instruction used for fine-tuning Falcon-7B is presented in Table 6. Also, a few-shot example for the PG task for inference from Chat-GPT and Falcon-7B is provided in table 7. Few-shot examples are selected from the development set for each task.

<human>:
[Context]: Being familiar with the working environment and able to intervene early is important for health care.
[Sentence]: Health care can be improved by being familiar with the working environment.
Is the Sentence supported by the Context above? Answer using ONLY yes or no:
<assistant>: [label]: yes

Table 6: Falcon-7B Fine-tuning Instruction Example

[Example 1]:
Context: I thought so, too.
Sentence: I thought you'd be surprised at me too.
Is the Sentence supported by the Context above? Answer using ONLY yes or no:
[label]: no

[Example 2]:
Context: I haven't been contacted by anybody.
Sentence: I have not been contacted.
Is the Sentence supported by the Context above? Answer using ONLY yes or no:
[label]: yes

[Example 3]:
Context: That was my general impression as well.
Sentence: I thought you'd be surprised at me too.
Is the Sentence supported by the Context above? Answer using ONLY yes or no:
[label]: no

[Example 4]:
Context: I said nothing of the kind.
Sentence: I never told you that before.
Is the Sentence supported by the Context above? Answer using ONLY yes or no:
[label]: yes

[Example 5]: the sample to be labeled...

Table 7: 4-Shot Chat-GPT Prompt Example

NIMZ at SemEval-2024 Task 9: Evaluating Methods in Solving Brainteasers Defying Commonsense

Zahra Rahimi, Mohammad Moein Shirzady, Zeinab Sadat Taghavi and Hossein Sameti
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
{zarahimi, mohammad.shirzady99, sameti}@sharif.edu,
zeinabtaghavi1377@gmail.com

Abstract

The goal and dream of the artificial intelligence field have long been the development of intelligent systems or agents that mimic human behavior and thinking. Creativity is an essential trait in humans that is closely related to lateral thinking. The remarkable advancements in Language Models have led to extensive research on question-answering and explicit and implicit reasoning involving vertical thinking. However, there is an increasing need to shift focus towards research and development of models that can think laterally. One must step outside the traditional frame of commonsense concepts in lateral thinking to conclude. Task 9 of SemEval-2024 is Brainteaser (Jiang et al., 2024), which requires lateral thinking to answer riddle-like multiple-choice questions. In our study, we assessed the performance of various models for the Brainteaser task. We achieved an overall accuracy of 75% for the Sentence Puzzle subtask and 66.7% for the Word Puzzle subtask. All the codes, along with the links to our saved models, are available on our GitHub[1].

1 Introduction

With recent advancements in deep learning and especially language models, extensive research has been conducted about reasoning in various natural language processing tasks, including question answering. These reasoning methods adopt vertical thinking. However, lateral thinking is another type often associated with creativity. In the 9th task of SemEval, Brainteaser (Jiang et al., 2024), a task of answering multiple-choice riddle-like questions is defined. To answer these questions, the model needs to employ lateral thinking. This method of thinking differs from vertical thinking in that the reasoning process is not linear. To arrive at a conclusion, one must examine the subject from a perspective beyond the usual conventional thinking paradigms (Waks, 1997). An example of a comparison between the two types of thinking is provided in Figure 2 in the Appendix. Lateral thinking demands a mind that is open, flexible, and creative. Equipping AI models with cognitive abilities such as lateral thinking can enhance problem-solving, adaptability, and coping with new situations and challenges.

In this work, we have evaluated the performance of three categories of models on answering brainteaser questions. We trained and evaluated two language models, BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019), the model presented in Yasunaga et al. (2021) (QA-GNN), and a T5 (Raffel et al., 2019) model for sentence puzzle and word puzzle subtasks. In the QA-GNN method, the ConceptNet knowledge graph (Speer et al., 2017) is used as the source of commonsense knowledge. Through the brainteaser task, we gained insights into two types of thinking - vertical and lateral. We also learned the significance of implementing lateral thinking in AI systems to bridge the gap between human and AI performance. Furthermore, this task piqued our interest in the captivating subject of creativity in artificial intelligence models. We achieved an overall accuracy of 75% and ranked 20th for the Sentence Puzzle subtask. For the Word Puzzle subtask, we ranked 19th and achieved an overall accuracy of 66.7%. All the codes, along with the links to our saved models, are available on our GitHub.

2 Background

The goal and dream of the artificial intelligence field has long been the development of intelligent systems or entities with human-like behavior and thinking. According to existing research, there are two types of thinking in humans: vertical and lateral. Most of the existing research focuses on vertical thinking. Vertical thinking involves a logi-

[1] https://github.com/z-rahimi-r/NIMZ-at-SemEval-Task-9-BRAINTEASER

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 148–154

cal and sequential approach, while lateral thinking requires creativity and flexibility to explore problems from unique and unconventional perspectives (Waks, 1997). The Brainteaser dataset (Jiang et al., 2023) contains 1100 riddle-like English questions requiring lateral thinking. The nature of questions often defies commonsense when approached with vertical thinking. The Brainteaser task includes two subtasks: sentence puzzle and word puzzle. The details of the dataset are presented in the Table 1.

Most research focuses on vertical thinking, using commonsense for implicit and explicit reasoning tasks such as commonsense question answering. Commonsense intelligence is intuitively reasoning about everyday situations and events, which requires knowledge of how the world works (Choi, 2022). In the task of commonsense question answering, two popular methods are fine-tuning language models and using graph neural network (GNN) models. In recent years, the use of knowledge graphs, the primary sources of commonsense knowledge, has increased. Commonsense knowledge stored in language model parameters is mainly descriptive and taxonomic knowledge, often explicitly stated in the language content that these models have been trained on (Hwang et al., 2021). The method presented in COMET (Bosselut et al., 2019) can be a means to teach language models other types of knowledge. The success of COMET can be attributed to the combination of neural and symbolic representations of knowledge, as well as the use of language to represent symbolic knowledge (Choi, 2022). The COMET model is fine-tuned on the ATOMIC knowledge graph (Hwang et al., 2021). This knowledge graph serves as a customized textbook for language models to learn commonsense knowledge and how the world works (Choi, 2022).

In the second popular category of methods, a knowledge graph is used as the complementary source of knowledge with the help of graph neural networks as the medium to harvest this knowledge (Feng et al., 2020; Wang et al., 2022; Zhang et al., 2022). One advantage of using graph neural networks is their interpretability. In QA-GNN (Yasunaga et al., 2021), the RoBERTa LM is used with graph neural networks. Each answer option is checked independently in their method to determine if it is the answer. For each answer option, a subgraph is extracted from the ConceptNet. This subgraph consists of the entities in question and the answer option, all the entities within two hops

	Sentence Puzzle	Word Puzzle
Train	507	395
Test	120	96

Table 1: Dataset Statistics

from question and answer entities on the ConceptNet graph, and the relations between them. In the presented method, the question and the answer option are concatenated and encoded using RoBERTa LM (Liu et al., 2019), then placed as a context node in the subgraph. Since some nodes in the subgraph are more related to the question and its answer, the RoBERTa LM is used to calculate a score for each node in the subgraph. This score is used as an additional feature to the node embeddings to increase the influence of more related entities. Training is done through the message-passing method. Finally, the score of each option being the answer is calculated and the answer option with the highest score will be the final answer to the question. The approach described in Zhang et al. (2023) is similar to QA-GNN but with one key difference. While QA-GNN evaluates each answer option independently using a local graph, this method also includes a global graph that allows for simultaneous evaluation and comparison of all answer options, leading to refined probabilities. Refining the probabilities of each answer option in this way can produce a more accurate result. They consider this method similar to how humans eliminate less likely options. The most similar available study to the Brainteaser task is Riddlesense (Lin et al., 2021), where a riddle dataset is presented. To solve the riddles, one needs advanced natural language understanding, commonsense, and counterfactual reasoning skills, which are complex cognitive processes. They have trained and evaluated several language models, GNN-based models, and text-to-text models on the Riddlesense dataset.

2.1 MCQA in LLMs

Inference from LLMs for multiple choice question answering is done using two methods: Multiple Choice Prompting (MCP) and Cloze Promting (CP) (Robinson et al., 2022). MCP involves presenting a question with several answer options to an LLM and asking it to select the most appropriate answer from the given choices. The other method, CP, involves creating a sentence or passage with a blank that the model needs to fill in with an appropriate

word or phrase. Robinson et al. (2022) criticizes using cloze-style prompts for evaluating LLMs, suggesting that this approach may not fully leverage these models' capabilities for MCQA tasks. However, the evaluation of LLMs with the MCP method has the problem that the order of presenting the options can change the final answer of the LLM. They have evaluated different LLMs, and based on the results, the model's size and providing examples (few-shot inference) to the language model can improve its performance and reduce the dependence of the final answer on the order of options.

2.2 How creative are LLMs?

Margaret Boden's criteria for creativity _novelty, value, and surprise_ are utilized to evaluate the creative capabilities of LLMs. Franceschelli and Musolesi (2023) discusses how much SOTA LLMs satisfy these criteria. LLMs can indeed produce valuable content, as evidenced by their impact and the quality of their outputs. The novelty of an idea or product is being dissimilar to existing examples, the reference of which can either be the person who comes up with it (psychological creativity) or the entire human history (historical creativity). Novelty in LLMs can occur accidentally or as a result of out-of-distribution production or careful prompts, and the degree of novelty is inherently limited by the models' design, focusing on probabilistic outputs based on historical data. The definition of surprise is how unexpected an idea is. Three types of surprise are defined: Combinatorial creativity, which is producing an unfamiliar combination of familiar ideas; Exploratory creativity, which is finding new and undiscovered solutions within the current style of thinking; and Transformational creativity, which is related to changing the current style of thinking. The autoregressive nature of LLMs makes the production of surprising content by them unlikely and only limited to combinatorial creativity, making truly surprising or transformational creativity challenging to achieve. True creativity requires self-awareness and self-evaluation capabilities, which current LLMs lack (Franceschelli and Musolesi, 2023).

3 System Overview

In this section, we will present the systems used to tackle the brainteaser task. The three main approaches in question-answering tasks are fine-tuning language models, graph neural networks, and text-to-text transformers. So, we decided to evaluate the performance of these models on the brainteaser task. Although the role of commonsense in this task is as a distractor (Jiang et al., 2023), we decided to evaluate the impact of using commonsense knowledge through ConceptNet knowledge graph and graph neural networks. While the answer may challenge commonsense in the Brainteaser questions, it does not violate it. All the models are trained for sentence puzzles and word puzzles separately. The general sketch for each type of system is presented in Figure 1.

3.1 Language models: BERT and RoBERTa

We trained and evaluated two language models, BERT-Base (Devlin et al., 2019) and RoBERTa-Large (Liu et al., 2019) on the Brainteaser dataset. The training was done on two different in-house splits of the training data, and the model with the best performance on the validation data was saved for final evaluation on the test set. During the training and inference phase for the two language models of BERT and RoBERTa, the probability of each option being the answer is checked separately. To do that, the question and the answer option are concatenated with the token [SEP] placed between them and given to the language model as input. The score of that option being the answer, is calculated using the output representation of the [CLS] token through a linear layer. Finally, the option that has the highest probability will be the answer to the question.

3.2 LM + GNN: QA-GNN

The QA-GNN model (Yasunaga et al., 2021) uses RoBERTa LM and graph neural networks for reasoning. The knowledge source used in this method is the ConceptNet knowledge graph (Speer et al., 2017). In this method, a separate subgraph is extracted for each answer option. The question and answer option are concatenated, and the resulting embedding from RoBERTa is used as a context node in the graph. This node is only connected to the entities belonging to the answer option and the question (it is not connected to other entities extracted from the knowledge graph). To train the QA-GNN model, pre-processing must be done on the dataset first. For each question and answer option pair, their entities and all of their neighbor entities up to two hops in the ConceptNet knowledge graph are extracted, along with the relations between them. Training is done through the message-

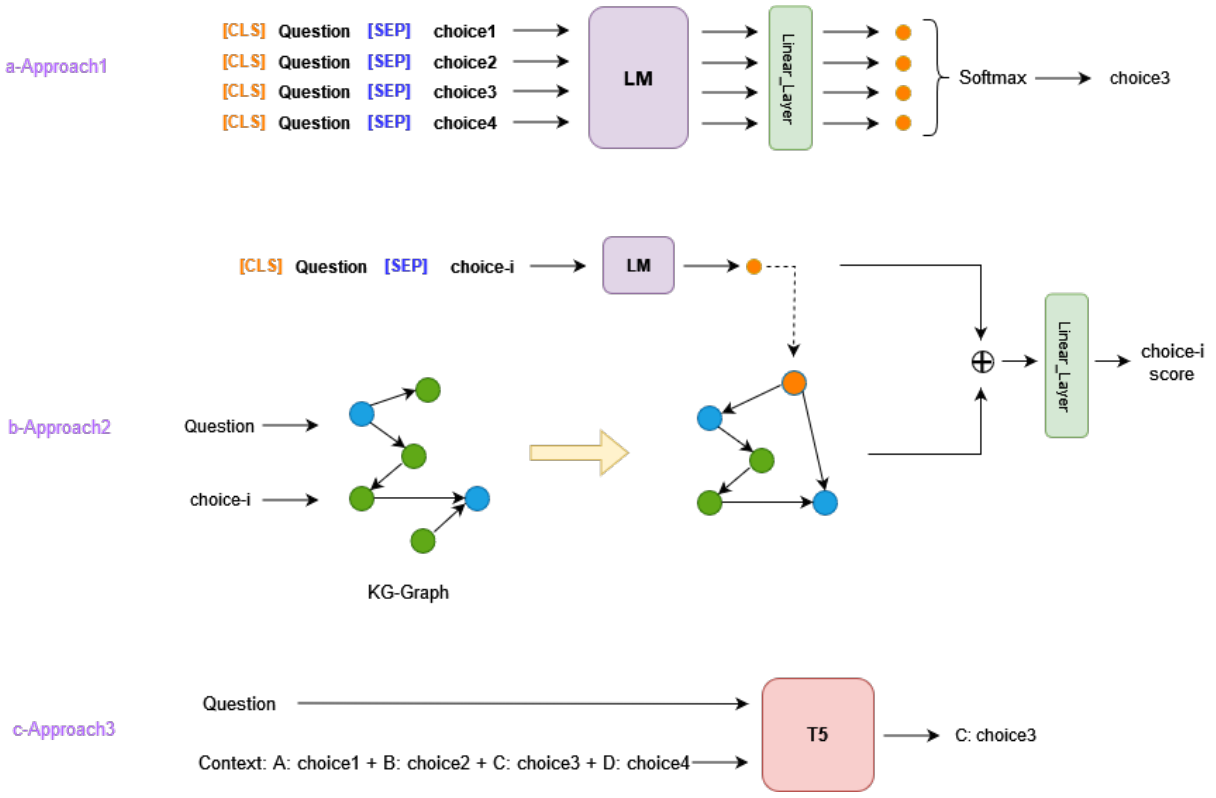


Figure 1: The three categories of methods evaluated for brainteaser task. **a-Approach1:** Fine-tuning LMs like BERT and RoBERTa; **b-Approach:** LM+GNN method, the blue circles are question and choice entities, and the green circles are extracted knowledge-graph entities; **c-Approach:** Fine-tuning a T5 model.

passing method. The score of each answer option, being the final answer, is calculated using the concatenation of the RoBERTa LM representation, context node representation learned through message-passing, and the pooled graph representation, through a linear layer. Finally, the option that has the highest score will be the answer to the question. The interested reader can refer to the original paper for more in-depth details.

3.3 Text-to-Text model

The third method we evaluated was the T5 text-to-text model (Raffel et al., 2019). In this method, the input question and the context, which includes all the options concatenated together, are passed to the T5 model as input. The answer will be in the form of a span extracted from the context, meaning the options. This model considers all options and makes a final decision, setting it apart from previous models.

4 Experimental Setup

We have trained and evaluated base and large sizes of BERT, RoBERTa, and T5 models using the Hugging-Face transformers library, with different hyperparameters to find the best setting. To train the QA-GNN model, we followed the procedure provided by the code available on the GitHub of Yasunaga et al. (2021). After preprocessing the Brainteaser dataset, the QA-GNN models were trained for 100 epochs with early-stopping. In the inference phase of the T5 model, in some cases, the extracted span was incomplete and did not include the letter of the answer option, in these cases the "none of above" option was selected. The code for the in-house train-dev split and the hyperparameters used for training the best-performing models are available in the notebooks on our GitHub[1].

4.1 Evaluation metrics

For each original question in the dataset, two additional adversarial variants are created: semantic reconstruction and contextual reconstruction. Semantic reconstruction rephrases the original question and does not change anything else. In contextual reconstruction, the context of the question does not change, but the surface form of the question and its answer options are changed. An example from

[1] https://github.com/z-rahimi-r/NIMZ-at-SemEval-Task-9-BRAINTEASER

		s_ori	s_sem	s_con	s_ori_sem	s_ori_sem_con	s_overall	w_ori	w_sem	w_con	w_ori_sem	w_ori_sem_con	w_overall
Baselines	Chat-GPT	0.608	0.593	0.679	0.507	0.397	0.627	0.561	0.524	0.518	0.439	0.292	0.535
	RoBERTa-Large	0.435	0.402	0.464	0.33	0.201	0.434	0.195	0.195	0.232	0.146	0.061	0.207
Eval.	BERT-Base	0.7	0.775	0.725	0.7	0.6	0.733	**0.7187**	**0.75**	0.531	**0.7187**	**0.4375**	**0.6666**
	QA-GNN	0.75	0.725	0.775	0.7	0.675	0.75	0.4375	0.4687	0.4375	0.4062	0.2187	0.4479
Post Eval.	RoBERTa-Large	**0.85**	**0.8**	**0.85**	**0.8**	**0.75**	**0.8333**	**0.7187**	0.6875	**0.5625**	0.625	0.375	0.6562
	T5-Large	0.55	0.625	0.525	0.5	0.275	0.5666	0.5937	0.5625	0.5312	0.4375	0.25	0.5625

Table 2: Results on Test set. The baselines are zero-shot results

the dataset is available in Table 3 in the Appendix. The purpose of designing these two variants is to test the robustness of the model. If the model has not memorized the content and is capable of lateral thinking, it will correctly answer these two adversarial variants of each question (Jiang et al., 2023). Models are evaluated using two accuracy metrics: instance-based accuracy metric and group-based accuracy metric. In instance-based accuracy, each question is evaluated separately. In group-based accuracy, a question is evaluated with its adversarial variants, and only if all three are answered correctly, it is scored One. Otherwise, it is scored Zero.

5 Results

We evaluated models from different categories on this task. Due to the riddle-like and unique nature of the questions, it was difficult for the models to generalize to new questions of the test set. We achieved an overall accuracy of 75% and ranked 20th for the Sentence Puzzle subtask. For the Word Puzzle subtask, we ranked 19th and achieved an overall accuracy of 66.7%. The QA-GNN model performed best for the sentence puzzle in the evaluation phase. Still, for the word puzzle, the BERT-base model had the best performance, and QA-GNN performed poorly, which could be due to the absence of reasoning paths on the knowledge graph between the concepts of the answer option and the question. The results of the two phases, evaluation and post-evaluation, are presented in the Table 2. Some wrongly predicted examples for the Word Puzzle subtask are presented in Table 4 in the Appendix.

6 Conclusion

In this study, we evaluated the performance of three main categories of popular methods in the question-answering task on the two subtasks of Sentence Puzzle and Word Puzzle of the SemEval- task 9 Brainteaser. We have achieved an overall accuracy of 75% for the Sentence Puzzle subtask and 66.7% for the Word Puzzle subtask. The nature of the Brainteaser questions is such that they challenge commonsense and require lateral thinking and intellectual creativity to be solved. Models other than LLMs tend to perform poorly in generalizing to new and different examples, especially when it comes to tasks that require creativity, such as puzzles and brainteasers. While LLMs tend to perform better, they still have limited capability when it comes to being creative. Regarding the suggestions for future work, we believe utilizing the chain-of-thought (Wu et al., 2023) method and teaching LLMs to reason step by step with the in-context-learning method can be effective. Another idea is to develop two modules for LLMs or AI agents. The first module will aid in the creative production of knowledge, while the second module will check the rationality of the produced knowledge and its consistency concerning the context of the desired problem. As mentioned earlier, the autoregressive nature of current LLMs and reliance on probabilistic solutions have limited their ability to produce creative content. So, there is a need to design new architectures and different training methods to overcome this limitation. This can be a helpful step towards enhancing creativity and lateral thinking in AI systems.

References

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *ArXiv*, abs/1911.11641.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Yejin Choi. 2022. The Curious Case of Commonsense Intelligence. *Daedalus*, 151(2):139–155.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

Giorgio Franceschelli and Mirco Musolesi. 2023. On the creativity of large language models.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1504–1515, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Joshua Robinson, Christopher Rytting, and David Wingate. 2022. Leveraging large language models for multiple choice question answering. *ArXiv*, abs/2210.12353.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6:245–255.

Ruijie Wang, Luca Rossetto, Michael Cochez, and Abraham Bernstein. 2022. Qagcn: A graph convolutional network-based multi-relation question answering system. Technical Report arxiv.2206, University of Zurich.

Dingjun Wu, Jing Zhang, and Xinmei Huang. 2023. Chain of thought prompting elicits knowledge augmentation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6519–6534, Toronto, Canada. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Qin Zhang, Shangsi Chen, Meng Fang, and Xiaojun Chen. 2023. Joint reasoning with knowledge subgraphs for multiple choice question answering. *Information Processing and Management*, 60(3):103297.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. *ArXiv*, abs/2201.08860.

Adv. Strategy	Question	Answers
Original	How could a cowboy ride into town on Friday, stay two days, and ride out on Wednesday?	**His horse is named Wednesday.** While in town, he stays in bed for two days. Friday and Saturday are holidays. None of the above.
Semantic Reconstruction	How could a cowboy come into town on Friday, stay two days, and then ride away on Wednesday?	**His horse is named Wednesday.** While in town, he stays in bed for two days. Friday and Saturday are holidays. None of the above.
Context Reconstruction	How can a pilot take off in Los Angeles on Tuesday, fly for 48 hours, and land in Tokyo on Tuesday?	**The pilot's airplane is named Tuesday.** He flies straight for 24h and flies quickly for hours left. There was a one-week long holiday. None of the above.

Table 3: A sentence-based lateral thinking puzzle and its adversarial variations from Brainteaser (Jiang et al., 2023)

Question	Choice List
What do you call a toothless bear?	A brown bear. **A polar bear.** **A gummy bear.** None of above.
What kind of birds always make noise?	**Humming bird.** **Hawk.** Owl. None of above.
What is the best key for a satisfying meal?	**A joykey.** **A turkey.** A hockey. None of above.
What lacks legs and feet but has toes?	Cabbages. **Tomatoes.** **Onions.** None of above.

Table 4: Examples of wrong predictions of Word Puzzle

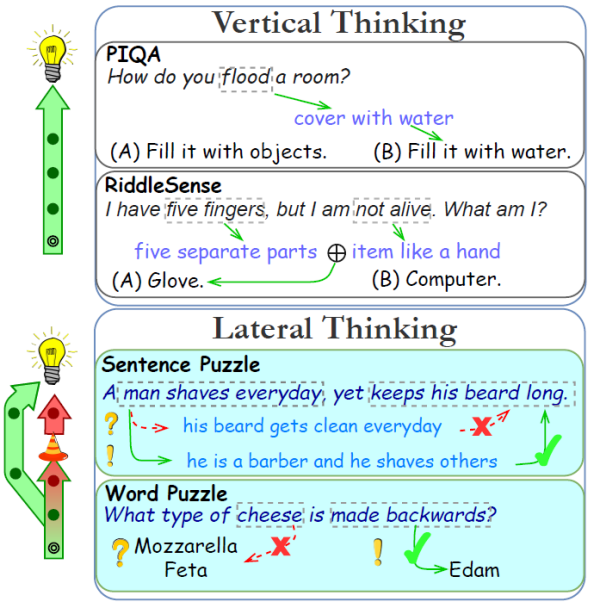


Figure 2: Comparing Vertical Thinking tasks (PIQA (Bisk et al., 2019) and RiddleSense (Lin et al., 2021)) to the BRAINTEASER lateral thinking task. (Jiang et al., 2023)

A Appendix

An example from the dataset is available in Table 3. Also, a few wrongly predicted examples for the Word Puzzle subtask are presented in Table 4. Figure 2 depicts a comparison of Vertical and Lateral thinking.

Mistral at SemEval-2024 Task 5: Mistral 7B for Argument Reasoning in Civil Procedure

Marco Siino
Department of Electrical, Electronic
and Computer Engineering
University of Catania
Italy
marco.siino@unipa.it

Abstract

At the SemEval-2024 Task 5, the organizers introduce a novel natural language processing challenge and corpus within the realm of the United States civil procedure. Every datum within the corpus comprises a comprehensive overview of a legal case, a specific inquiry associated with it, and a potential argument in support of a solution, supplemented with an in-depth rationale elucidating the applicability of the argument within the given context. Derived from a text designed for legal education purposes, this dataset presents a multifaceted benchmarking task for contemporary legal language models. Our manuscript delineates the approach we adopted for participation in this competition. Specifically, we detail the use of a Mistral 7B model to answer the questions provided. Our only and best submission reaches an F1-score equal to 0.5597 and an Accuracy of 0.5714, outperforming the task's baseline.

1 Introduction

The content of the Task 5 hosted at SemEval-2024 (Held and Habernal, 2024), was originally introduced in (Bongard et al., 2022).

Asserting a legal argument represents a fundamental proficiency necessary for aspiring legal professionals to acquire. This proficiency demands not only a comprehension of pertinent legal domains but also advanced reasoning skills, including the utilization of analogy-based arguments and the identification of implicit contradictions. Despite recent strides in establishing objective metrics for contemporary natural language processing (NLP) models across diverse facets of legal language comprehension, the absence of a sophisticated task addressing argumentative reasoning within legal contexts persists.

In this article, is discussed a novel task alongside a corresponding benchmark dataset. The introduction of a genuinely challenging task, sourced from legal educational resources, will serve to elucidate strengths and weaknesses inherent in contemporary legal transformer models, including but not limited to Legal-BERT (Chalkidis et al., 2020). Specifically, at the SemEval-2024 Task 5 is unveiled a novel, openly accessible legal dataset tailored for the binary text classification of issues within U.S. civil procedure. The primary objective is to ascertain whether a proposed solution to a given inquiry is deemed accurate or erroneous. The corpus draws inspiration from "The Glannon Guide To Civil Procedure" authored by Joseph Glannon (Glannon, 2023), which caters to law students by offering a comprehensive examination of fundamental U.S. civil procedure topics, inclusive of multiple-choice queries designed to assess reader comprehension.

Through the inception of this freshly minted corpus, the intent extends to scrutinizing the efficacy of various methodological approaches while establishing performance benchmarks.

To address these objectives, there is an ongoing demand for automated tools capable of extracting and categorizing data, facilitating the classification with recent NLP models. Recent advancements in the area of the machine and deep learning architectures have spurred heightened interest in Natural Language Processing (NLP). Substantial endeavours have been directed towards devising techniques for the automated identification and categorization of textual content accessible on the internet today. In the literature, to perform text classification tasks, several strategies have already been proposed. In the last fifteen years, some of the most successful strategies have been based on SVM (Colas and Brazdil, 2006; Croce et al., 2022), on Convolutional Neural Network (CNN) (Kim, 2014; Siino et al., 2021), on Graph Neural Network (GNN) (Lomonaco et al., 2022), on ensemble models (Miri et al., 2022; Siino et al., 2022) and, recently, on Transformers (Vaswani et al., 2017; Siino et al., 2022b).

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 155–162

Participants in SemEval-2024 Task 5 were tasked as follows. The task at hand involves evaluating the accuracy of an answer candidate provided in response to a question, accompanied by a brief introductory passage pertaining to the subject of the question. The objective is to ascertain whether the candidate answer is indeed incorrect or correct. To face with the task, we propose a Transformer-based approach which made use of Mistral 7B (Jiang et al., 2023). We used the model in a zero-shot setup described in the rest of this paper. Specifically, we prompted the latest pretrained version of Mistral with each sample in the dataset. Specifically, we provided a *candidate answer* to a *question*, asking the model if the answer to the legal question was correct or not. The model replied with a yes or no, eventually providing some further explanation.

The subsequent sections of this work are structured as follows: Section 2 offers background information on Task 5, held at SemEval-2024. In Section 3, we outline the approach introduced in this study. Section 4 delves into the specifics of the experimental setup employed to reproduce our findings. The outcomes of the official task and relevant discussions are presented in Section 5. Finally, Section 6 concludes our study and suggests avenues for future research.

We make all the code publicly available and reusable on GitHub[1].

2 Background

For the Task 5 at SemEval-2024 is proposed a legal corpus, publicly accessible for binary text classification tasks focusing on issues within U.S. civil procedure. The primary objective is to determine the correctness of solutions provided in response to specific questions. This corpus draws its content from "The Glannon Guide To Civil Procedure" authored by Joseph Glannon (Glannon, 2023), tailored for law students. The book encompasses fundamental U.S. civil procedure topics and includes multiple-choice questions aimed at evaluating reader comprehension.

Through collaboration with the author and publisher, task organizers secured permission to utilize the content of "The Glannon Guide To Civil Procedure" for constructing this dataset, which is freely available to the research community. The book comprises 25 chapters, each containing multiple-choice questions pertaining to a particular topic, prefaced by an introduction. Every question is followed by 3 to 5 answer candidates, among which one is deemed correct. These answer candidates serve as hypotheses, necessitating an examination of their respective prerequisites for accuracy. The correctness or incorrectness of an answer is subsequently expounded upon in the accompanying analysis.

The dataset construction process involved automated parsing of the book's content, leveraging its structured format to extract individual components of each instance (i.e., introduction, question, answers, and analysis). Additional parsing rules were employed to detect anomalies in the structure, such as instances where the same introduction was shared across multiple questions. However, certain sections of the book required manual extraction, particularly regarding the correctness of answer candidates, as this information was typically embedded within the free-text analysis section. The analysis segments were organized to address each answer candidate separately, classifying them as true or false. To achieve this, the organizers adopted a strategy of isolating the relevant aspects for each answer, despite the absence of explicit keywords or structural indicators guiding the segmentation process. Despite efforts to maintain consistency, some structural inconsistencies were noted throughout the dataset.

Two samples from provided datasets are available online[2] and reported in the Table 3 in the Appendix sectionA. In this case, the two samples contain the same introduction and the same question while providing different answers. Given the Introduction and the Question, the first answer (first row) is wrong, while the second one (second row) is correct.

The organizers adhere to the schedule for SemEval24, which means the following dates:

- Tasks announced (with sample data available): 17 July 2023
- Training data ready 4 September 2023
- Evaluation start 10 January 2024
- Evaluation end by by 31 January 2024
- Paper submission due 19 February 2024
- Notification to authors 18 March 2024

[1] https://github.com/marco-siino/SemEval2024/

[2] https://github.com/trusthlt/semeval24

- Camera ready due 01 April 2024
- SemEval workshop: June 16–21, 2024 (co-located with NAACL 2024 in Mexico City, Mexico)

3 System Overview

Even if it has already been proved that the Transformers are not necessarily the best option for any text classification task (Siino et al., 2022a), depending on the goal, some strategies like domain-specific fine-tuning (Sun et al., 2019; Van Thin et al., 2023), or data augmentation (Lomonaco et al., 2023; Mangione et al., 2022; Siino et al., 2024a) can be beneficial for the considered task.

So far, several Large Language Models (LLMs) have proved to be able to address a plethora of different NLP tasks. For example, in the recent literature, there has been mention of LLaMA, as presented by (Touvron et al., 2023). LLaMA stands out as a collection of publicly available Large Language Models (LLMs) that rival the capabilities of closed-source counterparts like GPT-3.

However, to address the Task 5 hosted at SemEval-2024 we made use of a zero-shot learning approach (Chen et al., 2023; Wahidur et al., 2024), making use of Mistral 7B (Jiang et al., 2023). Mistral 7B, a language model boasting 7 billion parameters, is engineered to excel in both performance and efficiency. In comparison to the leading open 13B model (Llama 2), Mistral 7B demonstrates superior performance across all assessed benchmarks. Moreover, it outperforms the leading publicly available 34B model (LLaMA 1) across various tasks involving code generation, mathematical operations, and reasoning. The model capitalizes on grouped-query attention (GQA) to expedite inference, complemented by sliding window attention (SWA) to effectively process sequences of varying lengths while minimizing inference costs. Additionally, a fine-tuned variant, Mistral 7B – Instruct, is tailored for adhering to instructions. This version, outperforms Llama 2 13B – chat model across both automated and human benchmarks.

The introduction of Mistral 7B Instruct underscores the ease with which the base model can be fine-tuned to achieve notable performance enhancements. Notably, this variant lacks any moderation mechanisms.

Our approach is few-shot (Littenberg-Tobias et al., 2022) and make use of the above-mentioned Mistral 7B. More specifically, given the task hosted at SemEval-2024, we asked the model: *"Is the Answer to the Question above True or False? Answer using ONLY True or False:"*. To this request, the model replied with one or more words - usually starting with a *true* or *false* - that we parsed to extract one of the two labels (i.e., 0 for false and 1 for true). For example, given the introduction:

> *"Defendant in denial. Cardozo is in an accident on Main Street with two other cars, driven by Hooper and Lopes. Cardozo brings a suit in federal court against Hooper and Lopes for his damages. Paragraph 21 of Cardozo's complaint alleges that Hooper had signaled before he turned onto Main Street. The police report on the accident states that, according to a bystander, Hooper had signaled before turning onto Main Street. Lopes, who was coming from Hooper's left, had no view of the right side of Hooper's car, and did not see whether he signaled or not. At the time an answer is due, Lopes's counsel has seen the police report, but has not yet been able to locate other witnesses to obtain their testimony. The most appropriate response for Lopes to Paragraph 21 of Cardozo's complaint would be to."*

The answer:

> *"state that he is without sufficient information to form a belief about the truth of the allegation."*

And our question:

> *Is the Answer to the Question above True or False? Answer using ONLY True or False:*

The model replied with:

> *true. lopes' answer could state that he lacks sufficient information to admit*

that we mapped into the binary label *1* corresponding to *true*.

We did not find any inconsistency in the outputs generated by Mistral along all the provided prompts. Specifically, we did not notice any variation in the behaviours of the model at different times of prompting. This leads us to the conclusion that given always the same input context (i.e.,

few-shot samples) during the prompt, the output provided is always consistent disregarding the time and the previous prompts provided. Finally, we collected all the predictions provided on the test set to into a JSON file with the required format to submit our predictions.

As noted in the recent study by (Siino et al., 2024b), the contribution of preprocessing for text classification tasks is generally not impactful when using Transformers. More specifically, the best combination of preprocessing strategies is not very different from doing no preprocessing at all in the case of Transformers. For these reasons, and to keep our system highly fast and computationally light, we have not performed any preprocessing on the text.

4 Experimental Setup

We implemented our model on Google Colab. The library we used come from HuggingFace and as already mentioned is Mistral 7B[3]. We employed the v0.2 iteration of Mistral 7B, which represents an enhanced version of the Mistral-7B-Instruct-v0.1 model. To harness the capabilities of instruction fine-tuning, prompts must be enclosed within [INST] and [/INST] tokens. Additionally, the initial instruction should commence with a sentence identifier. The next instructions should not. The assistant generation will be ended by the end-of-sentence token ID. We also imported the Llama library (Touvron et al., 2023) from *llama_cpp*. The library is fully described on GitHub[4]. The dataset provided for all the phases are available on the official competition page. We did not perform any additional fine-tuning on the model. To run the experiment, a T4 GPU from Google has been used. After the generation of predictions, we exported the results on the format required by the organizers. As already mentioned, all of our code is available on GitHub.

5 Results

Given the binary nature of the classification task, the organizers proposed F1 score and Accuracy as the two evaluation metrics to be considered for the final ranking. The F1 score is defined in the Equation 1. Where TP stands for the number of correctly predicted right answers, FP stands for the number of wrongly predicted right answers, and FN stands for the right answers wrongly predicted as wrong answers.

	F1	Accuracy
Mistral 7B	0.5597	0.5714

Table 1: The method's performance on the test set. In the table, the results obtained and shown on the official GitHub page are reported.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (1)$$

Given the previous definitions, the accuracy is defined as stated in the Equation 2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

In Table 1, we present the outcomes derived from our methodology. They are the same results publicly availble on the official final ranking shown on the official task page[5] and on CodaLab[6].

In the Table 2, the results obtained by the first three teams and by the last one, as showed on the official task page, are reported. Compared to the best performing models, our simple approach exhibits some room for improvements. However, it is worth notice that required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab. Finally, the proposed approach is able to outperform the baseline provided.

6 Conclusion

This paper presents the application of a Mistral 7B-model for addressing the Task 5 at SemEval-2024. For our submission, we decided to follow a zero-shot learning approach, employing as-is, an in-domain pre-trained Transformer. After several experiments, we found beneficial to build a prompt containing the question for the model. Then we provide as a prompt: the introduction, the question and an answer candidate. The model is asked to decide whether the candidate answer is correct or not. The task is challenging, and there is still opportunity for improvement, as can be noted looking at the final ranking. Possible alternative approaches include

[3] https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF

[4] https://github.com/ggerganov/llama.cpp

[5] https://github.com/trusthlt/semeval24

[6] https://codalab.lisn.upsaclay.fr/competitions/14817

TEAM NAME	F1	Accuracy
HW-TSC (1)	0.8231	0.8673
PoliToHFI (2)	0.7747	0.8265
SU-FMI (3)	0.7728	0.8367
lena.held (21)	0.4269	0.7449

Table 2: Comparing performance on the test set. In the table are shown the results obtained by the first three teams and by the last one. In parentheses is reported the position in the official final ranking.

utilizing the few-shot capabilities or also the use of other models like GPT and T5, eventually using further data, or directly integrating other samples from the training and from the development sets. Further improvements could be obtained with a fine-tuning and modelling the problem as a text classification task. Furthermore, given the interesting results recently provided on a plethora of tasks, also other few-shot learning (Wang et al., 2023; Maia et al., 2024; Siino et al., 2023; Meng et al., 2024) or data augmentation strategies (Muftie and Haris, 2023; Tapia-Téllez and Escalante, 2020; Siino and Tinnirello, 2023) could be employed to improve the results. Looking at the final ranking, our simple approach exhibits some room for improvements. However, it is worth notice that required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab.

Acknowledgments

We extend our gratitude to the anonymous reviewers for their insightful comments and valuable suggestions, which have significantly enhanced the clarity and presentation of this paper.

References

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The Legal Argument Reasoning Task in Civil Procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, UAE. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Shiming Chen, Ziming Hong, Wenjin Hou, Guo-Sen Xie, Yibing Song, Jian Zhao, Xinge You, Shuicheng Yan, and Ling Shao. 2023. Transzero++: Cross attribute-guided transformer for zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12844 – 12861.

Fabrice Colas and Pavel Brazdil. 2006. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Daniele Croce, Domenico Garlisi, and Marco Siino. 2022. An SVM ensemble approach to detect irony and stereotype spreaders on twitter. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2426–2432. CEUR-WS.org.

Joseph W Glannon. 2023. *Glannon guide to civil procedure: learning civil procedure through multiple-choice questions and analysis*. Aspen Publishing.

Lena Held and Ivan Habernal. 2024. SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Joshua Littenberg-Tobias, G. R. Marvez, Garron Hillaire, and Justin Reich. 2022. Comparing few-shot learning with GPT-3 to traditional machine learning approaches for classifying teacher simulation responses. In *AIED (2)*, volume 13356 of *Lecture Notes in Computer Science*, pages 471–474. Springer.

Francesco Lomonaco, Gregor Donabauer, and Marco Siino. 2022. COURAGE at checkthat!-2022: Harmful tweet detection using graph neural networks and ELECTRA. In *Proceedings of the Working Notes of*

CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022, volume 3180 of *CEUR Workshop Proceedings*, pages 573–583. CEUR-WS.org.

Francesco Lomonaco, Marco Siino, and Maurizio Tesconi. 2023. Text enrichment with japanese language to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2708–2716. CEUR-WS.org.

Beatriz Matias Santana Maia, Maria Clara Falcão Ribeiro de Assis, Leandro Muniz de Lima, Matheus Becali Rocha, Humberto Giuri Calente, Maria Luiza Armini Correa, Danielle Resende Camisasca, and Renato Antonio Krohling. 2024. Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer. *Expert Systems with Applications*, 241:122418.

Stefano Mangione, Marco Siino, and Giovanni Garbo. 2022. Improving irony and stereotype spreaders detection using data augmentation and convolutional neural network. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2585–2593. CEUR-WS.org.

Zong Meng, Zhaohui Zhang, Yang Guan, Jimeng Li, Lixiao Cao, Meng Zhu, Jingjing Fan, and Fengjie Fan. 2024. A hierarchical transformer-based adaptive metric and joint-learning network for few-shot rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(3).

Mohsen Miri, Mohammad Bagher Dowlatshahi, Amin Hashemi, Marjan Kuchaki Rafsanjani, Brij B Gupta, and W Alhalabi. 2022. Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*, 37(12):11319–11341.

Fuad Muftie and Muhammad Haris. 2023. Indobert based data augmentation for indonesian text classification. In *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, page 128 – 132.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2021. Detection of hate speech spreaders using convolutional neural networks. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 2126–2136. CEUR-WS.org.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2022a. Fake news spreaders detection: Sometimes attention is not all you need. *Information*, 13(9):426.

Marco Siino, Marco La Cascia, and Ilenia Tinnirello. 2022b. Mcrock at semeval-2022 task 4: Patronizing and condescending language detection using multi-channel cnn, hybrid lstm, distilbert and xlnet. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 409–417. Association for Computational Linguistics.

Marco Siino, Francesco Lomonaco, and Paolo Rosso. 2024a. Backtranslate what you are saying and i will tell who you are. *Expert Systems*, n/a(n/a):e13568.

Marco Siino, Maurizio Tesconi, and Ilenia Tinnirello. 2023. Profiling cryptocurrency influencers with few-shot learning using data augmentation and ELECTRA. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2772–2781. CEUR-WS.org.

Marco Siino and Ilenia Tinnirello. 2023. Xlnet with data augmentation to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2763–2771. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2022. T100: A modern classic ensemble to profile irony and stereotype spreaders. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2666–2674. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024b. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

José Medardo Tapia-Téllez and Hugo Jair Escalante. 2020. Data augmentation with transformers for text classification. In *Advances in Computational Intelligence*, pages 247–259, Cham. Springer International Publishing.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Dang Van Thin, Duong Ngoc Hao, and Ngan Luu-Thuy Nguyen. 2023. Vietnamese sentiment analysis:

An overview and comparative study of fine-tuning pretrained language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Rahman S. M. Wahidur, Ishmam Tashdeed, Manjit Kaur, and Heung-No Lee. 2024. Enhancing zero-shot crypto sentiment with fine-tuned language model and prompt engineering. *IEEE Access*, 12:10146 – 10159.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Trans. Circuits Syst. Video Technol.*, 33(12):7789–7802.

A Appendix

As stated in the background section, in this appendix are shown two samples from the provided datasets. The two samples in the Table 3 give an example of a wrong answer candidate (first row in the table) and an example of a correct answer candidate (second row in the table).

Introduction	Question	Answer Candidate	Label
"My students always get confused about the relationship between removal to federal court and personal jurisdiction. Suppose that a defendant is sued in Arizona and believes that she is not subject to personal jurisdiction there. Naturally, she should object to personal jurisdiction. [...] But generally the scope of personal jurisdiction in the federal court will be the same as that of the state court, because the Federal Rules require the federal court in most cases to conform to state limits on personal jurisdiction. Fed. R. Civ. P. 4(k)(1)(A). I've stumped a multitude of students on this point. Consider the following two cases to clarify the point."	*"7. A switch in time. Yasuda, from Oregon, sues Boyle, from Idaho, on a state law unfair competition claim, seeking $250,000 in damages. He sues in state court in Oregon. Ten days later (before an answer is due in state court), Boyle files a notice of removal in federal court. Five days after removing, Boyle answers the complaint, including in her answer an objection to personal jurisdiction. Boyle's objection to personal jurisdiction is"*	not waived by removal, but will be denied because the federal courts have power to exercise broader personal jurisdiction than the state courts.	0
"My students always get confused about the relationship between removal to federal court and personal jurisdiction. Suppose that a defendant is sued in Arizona and believes that she is not subject to personal jurisdiction there. Naturally, she should object to personal jurisdiction. [...] But generally the scope of personal jurisdiction in the federal court will be the same as that of the state court, because the Federal Rules require the federal court in most cases to conform to state limits on personal jurisdiction. Fed. R. Civ. P. 4(k)(1)(A). I've stumped a multitude of students on this point. Consider the following two cases to clarify the point."	*"7. A switch in time. Yasuda, from Oregon, sues Boyle, from Idaho, on a state law unfair competition claim, seeking $250,000 in damages. He sues in state court in Oregon. Ten days later (before an answer is due in state court), Boyle files a notice of removal in federal court. Five days after removing, Boyle answers the complaint, including in her answer an objection to personal jurisdiction. Boyle's objection to personal jurisdiction is"*	not waived by removal. The court should dismiss if there is no personal jurisdiction over Boyle in Oregon, even though the case was properly removed.	1

Table 3: Two different samples from the official dataset are provided. Together with the introduction, a question and a candidate answer the label is provided (i.e., 0 if the answer is incorrect, 1 if the answer is correct)

NCL-UoR at SemEval-2024 Task 8: Fine-tuning Large Language Models for Multigenerator, Multidomain, and Multilingual Machine-Generated Text Detection

Feng Xiong[1] and **Thanet Markchom**[2] and **Ziwei Zheng**[1] and **Subin Jung**[1] and **Varun Ojha**[1] and **Huizhi Liang**[1]
[1]School of Computing, Newcastle University, Newcastle upon Tyne, UK
[2]Department of Computer Science, University of Reading, Reading, UK
xf199912@163.com, t.markchom@pgr.reading.ac.uk, {z.zheng21, s.jung4, varun.ojha, huizhi.liang}@newcastle.ac.uk

Abstract

SemEval-2024 Task 8 introduces the challenge of identifying machine-generated texts from diverse Large Language Models (LLMs) in various languages and domains. The task comprises three subtasks: binary classification in monolingual and multilingual (Subtask A), multi-class classification (Subtask B), and mixed text detection (Subtask C). This paper focuses on Subtask A & B. To tackle this task, this paper proposes two methods: 1) using traditional machine learning (ML) with natural language preprocessing (NLP) for feature extraction, and 2) fine-tuning LLMs for text classification. For fine-tuning, we use the train datasets provided by the task organizers. The results show that transformer models like LoRA-RoBERTa and XLM-RoBERTa outperform traditional ML models, particularly in multilingual subtasks. However, traditional ML models performed better than transformer models for the monolingual task, demonstrating the importance of considering the specific characteristics of each subtask when selecting an appropriate approach.

1 Introduction

Large Language Models (LLMs) are sophisticated natural language processing (NLP) models extensively trained on vast textual datasets (Wang et al., 2023). These models demonstrate an impressive proficiency in generating human-like text based on the input they receive. However, using LLMs for generating texts has raised concerns about potential misuse, such as disseminating misinformation and disruptions in the education system (Wang et al., 2023). Thus, urgent development of automated systems to detect machine-generated texts is essential (Mitchell et al., 2023; Wang et al., 2023).

Recently, several LLMs have been developed such as ChatGPT[1] Brown et al. (2020), Cohere[2], Davinci[3], BLOOMZ[4] (Muennighoff et al., 2022), and Dolly[5] (Conover et al., 2023). The versatility of these models extends across various domains, such as news, social media, educational platforms, and academic contexts, in multiple languages not only English (Wang et al., 2023). This wide application poses a challenge in developing an automated system capable of detecting machine-generated texts from various generators, across multiple domains and languages.

To tackle this challenge, SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection (Wang et al., 2024) introduces the task of detecting machine-generated texts obtained from different LLMs, in various domains and languages. This task consists of three subtasks: Subtasks A, B, and C. Subtask A involves binary classification of text as either human-written or machine-generated, with two tracks: monolingual (English only) and multilingual. Subtask B focuses on multi-class classification of machine-generated text, aiming to identify the source of generation, whether human or a specific language model. Subtask C addresses the detection of human-machine mixed text, requiring the determination of the boundary where the transition from human-written to machine-generated occurs in a mixed text. This paper focuses on Subtasks A and B. To tackle these tasks, we propose two approaches: (1) classical machine learning, leveraging NLP techniques for feature extraction, and (2) fine-tuning LLMs for the classification of human-written and machine-generated texts.

2 Related Work

Researchers have employed a variety of methods and tools to detect AI-generated texts. Broadly,

[1]https://chat.openai.com/
[2]https://cohere.com
[3]https://platform.openai.com/docs/models/gpt-base
[4]https://huggingface.co/bigscience/bloomz
[5]https://huggingface.co/databricks/dolly-v2-12b

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 163–169

these approaches can be categorized into two main types: black-box and white-box detection methods (Tang et al., 2023). Black-box detection relies on API-level access to LLMs, utilizing textual samples from both human and machine sources to train classification models (Dugan et al., 2020). The study by Guo et al. (2023) integrated existing question-and-answer datasets and leveraged fine-tuning of pre-trained models to investigate the characteristics and similarities between human-generated and AI-generated texts.

As for white-box detection, Kirchenbauer et al. (2023) introduced a novel approach involving the embedding of watermarks in the outputs of LLMs to facilitate the detection of AI-generated text. Additionally, a variety of tools and methodologies, including XGBoost, decision trees, and transformer-based models, have been evaluated for their efficacy in detecting texts produced by AI (Zaitsu and Jin, 2023). These techniques incorporate multiple stylistic measurement features to differentiate between AI-generated and human-generated texts (Shijaku and Canhasi, 2023).

Specific tools and techniques in this domain include the GLTR tool developed by Gehrmann et al. (2019), which analyzes the usage of rare words in texts to distinguish between those generated by the GPT-2 model and human writers. The DetectGPT method posits that minor rewrites of LLM-generated texts tend to reduce the log probability under the model, a hypothesis that has been explored in depth (Mitchell et al., 2023). Furthermore, intrinsic dimension analysis, including methods like the Persistent Homology Dimension estimator (PHD), has been applied to distinguish between authentic texts and those generated artificially (Tulchinskii et al., 2023). Detectors specifically designed for certain LLMs, such as the GROVER detector for the GROVER model (Zellers et al., 2019) and the RoBERTa detector using the RoBERTa model (Liu et al., 2019), also play a significant role in this field.

In summary, the combination of statistical analysis with advanced language models is being employed by researchers to more effectively differentiate between content generated by humans and machines. The continuous evolution and refinement of these techniques reflect the dynamic nature of the field and the complexities involved in distinguishing between the increasingly nuanced outputs of LLMs and human-authored texts.

3 Methods

To tackle these tasks, we employ two distinct strategies. The first is classical machine learning, tailored for natural language preprocessing (NLP). The second approach involves transformer-based LLMs, with an emphasis on LoRA (Low-Rank Adaptation of Large Language Models) fine-tuning (Hu et al., 2021). We then enhance our results by integrating these methods through ensemble techniques.

3.1 Machine Learning Models

Our approach for textual data analysis in machine learning involves a concise yet comprehensive preprocessing pipeline. Initially, URLs and excess whitespace are removed from the text. Next, all punctuation is eliminated, focusing solely on alphanumeric characters. The text is further refined by excluding common stopwords and numeric characters. Emojis are decoded into text, providing additional context. Lemmatization standardizes words to their base forms, ensuring consistent analysis. Texts are then converted to lowercase for uniformity.

The final step involves using a *Term Frequency-Inverse Document Frequency* (TF-IDF), configured to handle a maximum of 8000 features and considering unigrams to trigrams. This vectorizer excludes terms appearing in less than 10 documents, balancing feature representation with computational efficiency. Furthermore, we enhance the feature set for machine learning by incorporating esteemed readability metrics such as the *Gunning fog index* (Scott, 2023) and *Flesch reading ease score* (Kincaid et al., 1975) into our text analysis, which assess the complexity and readability of the text respectively. This preprocessing strategy transforms raw text into a structured numerical format, ready for machine learning model analysis.

Expanding our feature extraction capabilities, we introduce additional dimensions of analysis including perplexity measures, sentiment analysis, document and error analysis, text vector features, the AI Feedback Query feature, and list lookup features. Perplexity measures assess text complexity through language models, offering insights into predictability. Sentiment analysis is deepened to reveal emotional tones and subjective nuances, providing a fuller understanding of the text's emotional landscape and authorial intent. Document and error analysis afford a detailed look at structure

and linguistic accuracy, enhancing content quality assessment. Text vector features, leveraging Sentence-BERT embeddings, enable sophisticated semantic content capture, facilitating nuanced thematic analysis. The AI Feedback Query feature is a binary response achieved through a structured inquiry where the AI model is presented with the text and asked to determine its generative source. List lookup features, examining elements like stop word frequency and special character use, offer stylistic and structural insights. Collectively, these advancements enable a comprehensive and detailed interpretation of textual data, significantly broadening our analytical capabilities by combining them.

In our study, we employed four distinct machine learning algorithms for both binary and multiclass classification tasks: Logistic Regression (LR), Multinomial Naive Bayes Classifier (MultinomialNB), eXtreme Gradient Boosting (XGBoost) (Chen and Guestrin, 2016) and Random Forest (RM).

- LR: A linear model used for classification tasks. It models the probability that a given input belongs to a certain class. Logistic Regression is particularly effective for binary classification due to its simplicity and efficiency in estimating probabilities.
- MultinomialNB: This algorithm is based on the Bayes theorem and is particularly suited for classification with discrete features (like word counts for text classification). It assumes independence between predictors and is highly scalable to large datasets.
- XGBoost: This is an efficient and scalable implementation of gradient-boosted decision trees. It is known for its performance and speed, especially in structured or tabular data, and can handle both binary and multi-class classification problems effectively.
- RF: A versatile ensemble learning method that builds multiple decision trees for classification or regression tasks. It improves accuracy by averaging or taking the mode of predictions from all trees, effectively reducing overfitting. Suitable for both binary and multiclass problems, it excels in handling large, high-dimensional datasets.

By integrating these algorithms, our approach leverages the strengths of linear modeling, probabilistic classification, and ensemble learning, aiming to enhance predictive accuracy and robustness across diverse classification scenarios.

3.2 XLM-RoBERTa

In our approach, we established XLM-RoBERTa[6] (Conneau et al., 2019) as the baseline model among transformer-based architectures. XLM-RoBERTa represents a multilingual adaptation of the original RoBERTa (Liu et al., 2019) model, specifically designed to understand and process a diverse range of languages. XLM-RoBERTa is pre-trained on a substantial dataset: 2.5TB of filtered CommonCrawl data (Zhang et al., 2020), encompassing text in 100 different languages. This extensive pre-training enables the model to capture nuanced language features and patterns across a broad linguistic spectrum, making it highly effective for tasks involving multiple languages. The use of such a diverse training dataset aids in achieving a robust understanding of various linguistic structures and vocabularies, which is crucial for accurate language processing and analysis in a multilingual context.

3.3 LoRA-RoBERTa

To improve the predictive performance of LLMs, we use LoRA for fine-tuning RoBERTa[7] model. LoRA is a technique enhancing the efficiency of fine-tuning large models with reduced memory consumption. It modifies the weight updates in neural networks using two smaller matrices derived through low-rank decomposition. These matrices adapt to new data while the original weights remain unchanged. The final output combines the original and adapted weights. In transformer models, LoRA is often applied to attention blocks for efficiency. The number of trainable parameters depends on the low-rank matrices' size, influenced by the rank and the original weight matrix's shape (Hu et al., 2021), as shown in Figure 1.

3.4 Majority Voting

The Majority Voting ensemble in this study combines the predictions of two transformer-based models: XLM-RoBERTa and LoRA-RoBERTa. The final prediction is determined by the majority vote of these two models, offers several advantages over a single-model approach. This technique, applicable in scenarios with N classifiers $(C_1, C_2, \ldots, C_N)$, determines the final out-

[6]https://huggingface.co/xlm-roberta-base
[7]https://huggingface.co/roberta-base

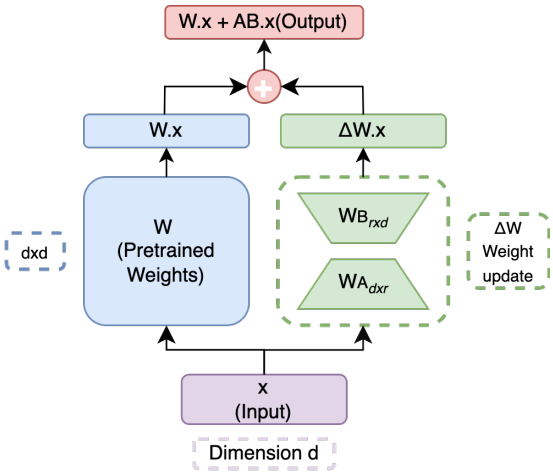


Figure 1: LoRA-based fine-tuning streamlines the process by freezing the original weights of LLMs and training a minimal number of parameters.

put $V(x)$ as the class receiving the most votes: $V(x) = \text{mode}\{C_1(x), C_2(x), \ldots, C_N(x)\}$. This method effectively reduces variance by balancing out individual model errors, leading to more stable predictions. Furthermore, it generally achieves higher accuracy due to the diverse perspectives of different models. Its robustness against overfitting is enhanced, as it combines various models' strengths, making it suitable for a wider range of data scenarios. The flexibility in model selection allows for a blend of different algorithms, each capturing unique data patterns, which contributes to better generalization on unseen data. Thus, majority voting stands out as a robust, accurate, and flexible approach in machine learning.

3.5 DistilmBERT

RoBERTa and XLM-RoBERTa are both powerful but computationally expensive. Therefore, we investigate an alternative model that is more computationally efficient, aiming to compare its performance against these models. We adopted *DistilBERT base multilingual cased*[8] (DistilmBERT) (Sanh et al., 2019), a distilled version of the BERT base multilingual model. It was pretrained on the concatenation of Wikipedia in 104 different languages. DistilmBERT consists of 6 layers, each with 768 dimensions and 12 attention heads, totaling 134 million parameters. This configuration balances model efficiency while retaining significant representational power Sanh et al. (2019).

[8]https://huggingface.co/distilbert-base-multilingual-cased

4 Experiments

In our study, subtask A focuses on distinguishing between human-written (label 0) and machine-generated text (label 1), offered in both monolingual (119,757 train, 5,000 dev, 34,272 test) and multilingual versions (172,417 train, 4,000 dev, 42,378 test), across various sources and languages are given in Table 1. Subtask B, with 71,027 train, 3,000 dev, and 18,000 test, goes further by identifying the specific model (including ChatGPT, Cohere, DaVinci, BloomZ, and Dolly) that generated the text, or if it's human-generated. Both tasks utilize datasets with an identifier, label, text content, model name, and source, focusing on the nuanced classification of texts.

Subtask	#Train	#Dev	#Test
A - Monolingual	119,757	5,000	34,272
A - Multilingual	172,417	4,000	42,378
B	71,027	3,000	18,000

Table 1: Dataset for text classification subtasks

4.1 Parameter Settings

In our experimentation, hyperparameter settings varied between classical machine learning models and LLMs. For the classical machine learning models, we adhered to default parameter settings during training. This approach simplifies the process and relies on the general applicability of these preset parameters.

In contrast, for LLMs, specific hyperparameters were carefully chosen. When training the XLM-RoBERTa baseline model, we set the batch size to 16 and the learning rate to $2.0e-5$ with the model being trained for 3 epochs. This configuration ensures efficient handling of data and optimal learning speed. For fine-tuning the LoRA-RoBERTa base model, the learning rate was adjusted to $1.0e-3$ over 5 epochs, a setting conducive to the specific demands of fine-tuning.

Furthermore, we employed configuration for the LoRA fine-tuning, defined with the following parameters: *task_type* set to *SEQ_CLS* indicating a sequence classification task, *r* (rank of the low-rank matrices) set to 4, *lora_alpha* (scaling factor for learning rate) at 32, *lora_dropout* to manage overfitting set at 0.01, and *target_modules* focused on the *query* module. These configurations are critical in guiding the fine-tuning process, ensuring that the

Method	Subtask A - Monolingual		Subtask A - Multilingual		Subtask B	
	Dev	Test	Dev	Test	Dev	Test
LR	0.673	0.764	0.473	0.721	0.251	0.393
MultinomialNB	0.555	**0.832**	0.483	0.717	0.435	0.511
XGBoost	0.692	0.800	0.515	0.738	0.540	0.545
RF	0.650	0.825	-	-	0.471	0.524
XLM-RoBERTa	**0.783**	0.717	0.679	**0.875**	**0.735**	0.600
LoRA-RoBERTa	**0.783**	0.811	0.726	0.672	**0.735**	**0.699**
Majority voting	0.735	0.828	**0.728**	0.862	0.717	0.602
DistilmBERT	0.702	0.730	0.670	0.810	0.629	0.619

Table 2: Performance comparison of ML and transformer models on text classification subtasks

adjustments to the model are precisely tailored to enhance performance on the specified task.

As for DistilmBERT, the maximum length of input sequences was set to 512. The AdamW optimizer was employed for training with a learning rate set to $1.0e-4$ and a batch size of 20. This model was trained for 5 epochs.

4.2 Results and Discussions

In our experiments, we evaluated various models on three distinct subtasks: Subtask A - Monolingual, Subtask A - Multilingual, and Subtask B. Each subtask involved both development (Dev) and test phases. The models tested included traditional machine learning algorithms - LR, MultinomialNB, XGBoost and RF - as well as advanced transformer-based models like XLM-RoBERTa, LoRA-RoBERTa, and DistilmBERT. However, due to the complexity of RF and time constraints, experiments on this approach for Subtask A - Multilingual are still ongoing, we plan to report the results in future work. Additionally, we employed a majority voting ensemble method combining XLM-RoBERTa and LoRA-RoBERTa.

The results, detailed in Table 2, reveal significant variations in model performance across the subtasks, highlighting the strengths and weaknesses of each model. One notable observation is the large performance gap between the dev and test sets for some ML approaches. This discrepancy could be attributed to several factors, such as overfitting, differences in data distribution between the dev and test sets, or the limited complexity of some ML models in capturing the intricacies of the task. Further investigation and error analysis are necessary to fully understand and address these issues.

Subtask A - Monolingual In the monolingual Subtask A, MultinomialNB emerged as a strong performer with the highest test score of 0.832. RF and XGBoost also showed robust performance with test scores of 0.825 and 0.800, respectively. The success of these ML models in the monolingual setting suggests that they can effectively capture relevant features and patterns when dealing with a single language. However, their performance on the dev set was notably lower, indicating potential overfitting or limitations in generalizing to unseen data. Among the transformers, LoRA-RoBERTa was notable with a test score of 0.811, outperforming XLM-RoBERTa, which scored 0.717. DistilmBERT, while not leading, still demonstrated a commendable test score of 0.730, indicating its effectiveness in monolingual contexts. The performance of transformer models in this subtask highlights their ability to capture complex language representations and generalize well to new data.

Subtask A - Multilingual In the challenging multilingual Subtask A, XLM-RoBERTa excelled with the highest test score of 0.875. The Majority Voting ensemble was also highly effective, achieving a test score of 0.862. These results demonstrate the strength of transformer models in handling diverse language inputs and their ability to learn language-agnostic representations. DistilmBERT, with a test score of 0.810, also showed notable effectiveness in multilingual text classification, outperforming traditional models and reflecting its potential in handling complex, diverse language data.

Subtask B In Subtask B, LoRA-RoBERTa led with a Test score of 0.699, follwed by DistilmBERT, achieving a test score of 0.619 and XLM-RoBERTa with 0.600. The strong performance

of transformer models in this subtask underscores their versatility and adaptability across different text classification scenarios. Among the traditional models, XGBoost was the most effective, with a test score of 0.545. However, the performance gap between ML models and transformers in Subtask B suggests that the latter are better equipped to handle the specific challenges and complexities of this task.

At the model level, we observed that ML models often struggled with handling rare or out-of-vocabulary words, leading to misclassifications. Transformer models, on the other hand, showed better resilience to such challenges, likely due to their subword tokenization and ability to capture broader context. However, transformers sometimes struggled with very short or noisy inputs, indicating room for improvement in their robustness.

5 Conclusions

The results showed that transformer models, particularly LoRA-RoBERTa and XLM-RoBERTa, performed exceptionally well in most text classification tasks. DistilmBERT represented a more streamlined transformer approach and was also proven to be efficient, especially in multilingual task. Contrary to popular belief, traditional ML models such as MultinomialNB and XGBoost can outperform transformers in monolingual tasks. These findings highlight the importance of carefully considering the characteristics of the task and the trade-offs between model complexity and performance when selecting an appropriate approach.

Our results contribute to the understanding of model selection strategies for text classification and emphasize the need for a nuanced approach that takes into account the specific demands of each subtask. Future research could explore the development of hybrid models that combine the strengths of traditional ML techniques and transformer architectures, as well as the design of more efficient and lightweight transformer models for resource-constrained environments. These findings reflected the dynamic nature of NLP tools and the importance of selecting models based on the specific requirements of the task.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. Roft: A tool for evaluating human detection of machine-generated text. *arXiv preprint arXiv:2010.03070*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Brian Scott. 2023. The gunning's fog index (or fog) readability formula.

Rexhep Shijaku and Ercan Canhasi. 2023. Chatgpt generated text detection. *Publisher: Unpublished*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Serguei Barannikov, Irina Piontkovskaya, Sergey Nikolenko, and Evgeny Burnaev. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts. *arXiv preprint arXiv:2306.04723*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, et al. 2023. M4: Multigenerator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Wataru Zaitsu and Mingzhe Jin. 2023. Distinguishing chatgpt (-3.5,-4)-generated and human-written papers through japanese stylometric analysis. *arXiv preprint arXiv:2304.05534*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Hao Zhang, Jae Ro, and Richard Sproat. 2020. Semi-supervised url segmentation with recurrent neural networks pre-trained on knowledge graph entities. *arXiv preprint arXiv:2011.03138*.

iML at SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials with LLM Based Ensemble Inferencing

Abbas Akkasi[1], Adnan Khan[1], Mai A. Shaaban[2], Majid Komeili[1], Mohammad Yaqub[2],

[1]School of Computer Science Carleton University, Ottawa, Canada
[2] Mohamed bin Zayed University of Artificial Intelligence Abu Dhabi, UAE
Correspondence: abbasakkasi@cunet.carleton.ca

Abstract

The task of textual entailment holds significant importance when dealing with clinical data, as it serves as a foundational component for extracting and synthesizing medical information from vast amounts of unstructured text.

To investigate the consistency with which Natural Language Inference (NLI) models capture semantic phenomena critical for intricate inference within clinical NLI contexts, SemEval−2024 has organized a shared task focused on NLI for Clinical Trials (NLI4CT). This task provides participants with a dataset annotated by humans for the purpose of model training and requires the submission of the results on test data for evaluation. We engaged in this shared task2 at SemEval−2024, employing a diverse set of solutions, with a particular emphasis on leveraging a Large Language Model (LLM) based zero-shot inference approach to address the challenge.

1 Introduction

Clinical NLI is a specialized application of Natural Language Processing (NLP) that focuses on understanding and inferring information from text within the healthcare domain. It involves analyzing and drawing conclusions from clinical narratives, such as electronic health records (EHRs), doctor's notes, medical transcripts, clinical trials and other forms of medical documentation (Percha et al., 2022). The goal of clinical NLI is to determine the logical relationship between premises and hypotheses (conclusions) in clinical text. By inferring information from clinical text, NLI can assist healthcare providers in making informed decisions by providing evidence-based recommendations and alerts. In addition, clinical NLI can be used to identify patient cohorts for clinical trials or research studies by inferring patient eligibility based on inclusion and exclusion criteria mentioned in clinical records. Applications of clinical NLI are not limited to the ones mentioned and there are lots of other usages in which clinical NLI can be useful (Percha et al., 2021). NLI for clinical trials faces unique challenges due to the complexity of medical language, the need for domain-specific knowledge, and the sensitivity and privacy concerns associated with health data. However, advancements in NLP and specifically Large Language Models (LLMs) are continuously improving the accuracy and applicability of clinical NLI, making it an increasingly valuable tool in the healthcare industry.

To foster collaboration and dissemination of novel insights within this field, SemEval 2024 (Jullien et al., 2024) has established a shared task exclusively devoted to clinical NLI. A publicly accessible dataset, annotated by humans, has been made available to facilitate the comparison of solutions proposed by different researchers.

To address the challenge, we developed an ensemble-oriented solution that combines various Large Language Models (LLMs) based models within the framework of prompting and fine-tuned classification. Our primary goals were to first understand the comparative performance of generative models versus classification models. Subsequently, we explored whether the use of automatic summarization models to condense the premises would influence the efficacy of both classifiers and generative models. Ultimately, our approach sought to facilitate synergistic interactions among the different models, leveraging their respective strengths to mitigate individual inference limitations.

Nevertheless, despite conducting a variety of experiments that involved combining summarization, fine-tuning classifiers, prompting, and more, the results demonstrated a clear superiority of generative models in comparison to the others, even when used independently.

The remainder of this paper is organized as follows: Section 2 provides a brief review of related

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 170–174

work. The proposed model and its constituent modules are detailed in Section 3. Sections 4 and 5 discuss the experiments conducted and the corresponding results. Finally, we conclude the paper in Section 6.

2 Past Work

Recent literature underscores the need for sophisticated models that can accurately capture the semantics of clinical narratives and support reasoning in line with medical knowledge. Jullien et al. (2023), introduced a shared task on NLI for clinical trials (NLI4CT), providing a dataset of annotated clinical trials and inviting researchers to develop models to tackle the associated challenges. The shared task comprises two sub-tasks: Textual Entailment and Evidence Retrieval, each designed to advance the state of NLI systems within the clinical domain.

Zhou et al. (2023), took part in the NLI4CT-2023 challenge, proposing a model that utilizes both sentence-level and token-level encoding to address the task at hand. Furthermore, they enhanced the model's overall performance by employing general (T5-based model) and domain-specific (SciFive) pre-trained LLMs.

Kanakarajan and Sankarasubbu (2023), conducted an evaluation of several instruction-tuned Large Language Models (LLMs) in a zero-shot setting and fine-tuned the best-performing instruction-tuned model (T5 family models). Their findings suggest that instruction-tuned models yield better results for datasets with limited training samples. Additionally, they explored the impact of various prompts on the overall performance of the model. (Vladika and Matthes, 2023) and (Chen et al., 2023), both created a model based on an ensemble approach that combines various fine-tuned iterations of biomedical LLMs. These models are designed to extract evidence from clinical trial report premises to support textual entailment in specific statements. Wang et al. (2023), developed a system that utilizes prompts created by humans to gather information from statements, section titles, and clinical trials. They then fine-tune pre-trained language models on these prompted sentences, training the models to identify the inferential connections between the statements and the clinical trials. Pahwa and Pahwa (2023), characterized the NLI task as a form of text pair classification and utilized the GPT-3 model to classify samples within the framework of few-shot prompting. This approach takes advantage of the semantic similarity between text samples and the examples provided for in-context learning.

Dias et al. (2023), employed supervised contrastive learning to enhance the sentence pair representations in the Biomed RoBERTa model. They then fine-tuned a linear classifier built upon these improved representations to identify evidence and execute textual entailment classification for sentence pairs.

Vassileva et al. (2023), introduced a two-tiered system to address the sub-tasks of NLI4CT-2023. Initially, the system employs a BERT-based classifier, supplemented by contextual data augmentation, to categorize evidence-statement pairs as relevant or irrelevant. Subsequently, leveraging the relevant segments of the clinical trial identified in the first stage, the system applies another BERT-based classifier to ascertain whether the relationship between the elements is one of entailment or contradiction.

Volosincu et al. (2023), illustrated that a transformer model pre-trained on biomedical data for the task of entailment relation in NLI4CT-2023 does not automatically outperform traditional approaches like CNNs. Nonetheless, their model exceeded the baseline system's performance and provided meaningful directions for future research on how the model's architecture can be developed further.

3 Proposed Model

In tackling the NLI4CT task, our approach involved the construction of an ensemble model that integrates the judgments of multiple distinct decision-makers. These decision-makers differ concerning the nature of input data they process, the foundational models they employ, and the methodologies they adopt for label determination. Figure 1 provides a comprehensive illustration of the proposed solution. Components of the ensemble pool were developed within the frameworks of classification or prompting, utilizing LLMs. For classification tasks, SciFive (Zhou et al., 2023) was selected as the base model due to its exemplary performance in the NLI4CT-2023 task. To enhance the models' ability to assimilate information from the input data, we employed both extractive and abstractive summarization techniques. The abstractive summarization was conducted using the T5-large model (Raffel et al., 2020) to condense the premises. For

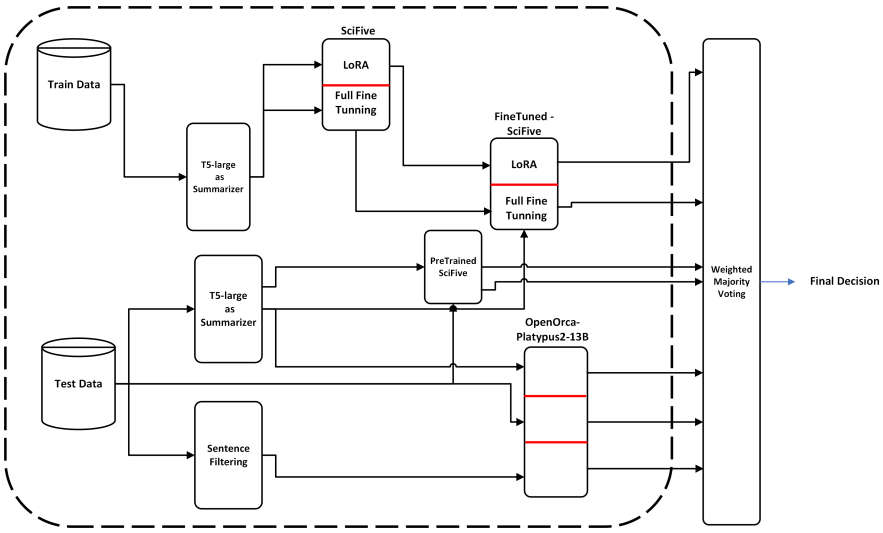


Figure 1: Ensemble Model Proposed

extractive summarization, the premises were initially segmented into individual sentences, after which those exhibiting lower semantic similarity to the hypothesis were excluded.

The pre-trained SciFive model ingests the text summarized by T5 to generate the initial component of the ensemble pool. Subsequently, this model undergoes fine-tuning through two distinct methodologies utilizing the summarized data: comprehensive fine-tuning and parameter-efficient fine-tuning, the latter of which is facilitated by employing LoRA (Hu et al., 2021) to produce subsequent members of the ensemble pool.

The remaining decision-makers within the ensemble are derived by prompting generative LLM[1] in a zero-shot inference context, utilizing both the original input data and variously summarized inputs. The specific prompt employed for the model is delineated in Listing 1.

```
# For Type="Comparison"
prompt = f''' Assess the logical
    relationship between two clinical
    trial descriptions (Primary Trial (
    PT), Secondary Trial: (ST)) as
    premises and the hypothesis given
    below.
     Return 'Entailment' if the premises
     logically imply the hypothesis, and
     'Contradiction' if the hypothesis
     conflicts with the information in
     the premises.
     Primary Trial (PT) : {PE}
     Secondary Trial (ST): {SE}
     hypothesis: {hypothesis}
'''
# For Type="Single"
prompt = f'''Evaluate the logical
    relationship between the clinical
    trial premise (PE) and the
    hypothesis given below.
     Return 'Entailment' if the premise
    logically implies the hypothesis,
    and 'Contradiction' if the
    hypothesis conflicts with the
    information in the premise.
Clinical Trial (PE): {PE}
hypothesis: {hypothesis}
'''
```

Listing 1: Prompt Template Used.

Ultimately, the final decision for the test samples were made using a weighted majority voting approach. The performance of models on *practice_test* set were used for the combination process.

4 Experiments

We have conducted our experiments utilizing the dataset provided by the task's organizers that is explained in Section 4.1. For the models based on prompting, we utilized only the test and practice_test datasets, whereas the training data was employed exclusively for fine-tuning the classification-based models. Beyond experimenting with models within our ensemble framework, we also explored the integration of results from fine-tuned classification models as a form of external knowledge within the context of prompting. The efficacy of all models is evaluated using three metrics: Macro

[1]OpenOrca-Platypus2-13B, which is an autoregressive language model that utilizes the Lllama 2 transformer architecture. It is tailored for a variety of general-use applications, including chat, text generation, and code generation. This model has undergone training with a diverse mix of datasets, focusing on STEM and logic-based content, and it incorporates a carefully selected portion of data from the GPT-4 dataset within the OpenOrca collection.

F1 Score, Faithfulness, and Consistency, each of which is briefly described in Section 4.2.

4.1 Dataset

The corpus presented for analysis encompasses training, development, practice_test, and test datasets, each containing a distinct number of samples. Table 1 displays the quantity of samples for each dataset. The content of each sample, including statements and evidence, has been reconstructed by a collaborative effort of clinical domain experts, clinical trial organizers, and research oncologists associated with the Cancer Research UK Manchester Institute and the Digital Experimental Cancer Medicine Team.

Split	#Samples	#Entailment	#Contradiction
Train	1700	850	850
Practice_test	2142	730	1412
Development	200	100	100
Test	5500	1841	3659

Table 1: Overview of Dataset Splits: Distribution of Samples, Entailment, and Contradiction Labels

4.2 Evaluation

In assessing system performance, the organizers, in conjunction with the macro F1 score, opted to examine model efficacy on a contrast dataset comprising statements with interventions. The comprehensive ranking of the systems is determined by the mean of two novel metrics: Faithfulness (as defined in Equation. 1) and Consistency (as defined in Equation. 2), across all types of interventions.

$$FaithFulness = \frac{1}{N}\sum_{1}^{N} |f(y_i) - f(x_i)| \quad (1)$$

where $x_i \in C : Label(x_i) \neq Label(y_i), f(y_i) = Label(y_i)$.

$$Consistency = \frac{1}{N}\sum_{1}^{N} 1 - |f(y_i) - f(x_i)| \quad (2)$$

where $x_i \in C : Label(x_i) = Label(y_i)$. Faithfulness quantifies the degree to which a system reaches an accurate prediction based on the correct rationale. While, Consistency measures the degree to which a system yields identical outputs for semantically equivalent queries. The results obtained during the experimental trials are presented in the subsequent section.

5 Results

The performance result of individual models within the ensemble, as applied on both practice_test and test datasets, are illustrated in Table 2.

The proposed model exhibits faithfulness and consistency scores of 28% and 52%, respectively, suggesting a necessity for more robust models to effectively manage clinical trials involving diverse data types. The findings reveal that the proposed overall model performs similarly to the generative model in the prompting context. This similarity underscores the considerable potential of generative LLMs. These models can achieve better performance when instruction tuning is applied with domain-specific data. Additionally, using classification results as external knowledge for the prompting model showed minimal impact. Moreover, the use of extractive summarization yielded the lowest results, aligning with our expectations. This approach, which focuses on the similarity between individual sentences and the statement, can lead to a loss of comprehension of the entirety of the premises.

6 Conclusion

In conclusion, our participation in NLI4CT-2024 involved proposing an ensemble approach that incorporated multiple decision-makers, with two Large Language Models (LLMs) serving as foundational models. We explored various data preparation techniques, including abstractive summarization and similarity-based sentence filtering, for use in both prompting and classification contexts. The comparable performance of the prompt-based model to the overall ensemble model, coupled with its significant outperformance of the classification models, underscores the substantial potential of pretrained generative foundation models in solving similar problems. We posit that the application of instruction tuning and the incorporation of domain-specific data could markedly enhance the results.

7 Acknowledgments

We extend our sincere gratitude to Dana Osama and Anees Hashmi for their valuable cooperation and contributions to this work. In addition, this work received support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Digital Alliance of Canada, to whom the authors extend their gratitude.

	Practice_Test							
	M1	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Score	66.66	72.64	66.89	72.65	68.12	60.66	69.12	**72.65**
	Test							
	M1	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**
Score	66.84	65.37	66.30	69.61	66.36	52.95	66.07	**70.27**

Table 2: Performance comparison in terms of F1-score on practice test and test Datasets: M1: Pretrained SciFive, M2: Full Fine-tuned SciFive (Summarized Data), M3: Fine-tuned SciFive (LoRA and Summarized Data), M4: Prompting, M5: Prompting with Summarized Data, M6: Prompting with Filtered Sentences, M7: SciFive Results as External Knowledge for Prompting, M8: Ensemble Method

References

Chao-Yi Chen, Kao-Yuan Tien, Yuan-Hao Cheng, and Lung-Hao Lee. 2023. Ncuee-nlp at semeval-2023 task 7: Ensemble biomedical linkbert transformers in multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 776–781.

Abel Corrêa Dias, Filipe Dias, Higor Moreira, Viviane Moreira, and João Luiz Comba. 2023. Team inf-ufrgs at semeval-2023 task 7: Supervised contrastive learning for pair-level sentence classification and evidence retrieval. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 700–706.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama ai research at semeval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003.

Bhavish Pahwa and Bhavika Pahwa. 2023. Bphigh at semeval-2023 task 7: Can fine-tuned cross-encoders outperform gpt-3.5 in nli tasks on clinical trial data? In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1936–1944.

Bethany Percha, Kereeti Pisapati, Cynthia Gao, and Hank Schmidt. 2021. Natural language inference for clinical registry curation. *medRxiv*, pages 2021–06.

Bethany Percha, Kereeti Pisapati, Cynthia Gao, and Hank Schmidt. 2022. Natural language inference for curation of structured clinical registries from unstructured text. *Journal of the American Medical Informatics Association*, 29(1):97–108.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sylvia Vassileva, Georgi Grazhdanski, Svetla Boytcheva, and Ivan Koychev. 2023. Fmi-su at semeval-2023 task 7: Two-level entailment classification of clinical trials enhanced by contextual data augmentation. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1454–1462.

Juraj Vladika and Florian Matthes. 2023. Sebis at semeval-2023 task 7: A joint system for natural language inference and evidence retrieval from clinical trial reports. *arXiv preprint arXiv:2304.13180*.

Mihai Volosincu, Cosmin Lupu, Diana Trandabat, and Daniela Gifu. 2023. Fii smart at semeval 2023 task7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 212–220.

Weiqi Wang, Baixuan Xu, Tianqing Fang, Lirong Zhang, and Yangqiu Song. 2023. Knowcomp at semeval-2023 task 7: Fine-tuning pre-trained language models for clinical trial entailment identification. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1–9.

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. Thifly research at semeval-2023 task 7: A multi-granularity system for ctr-based textual entailment and evidence retrieval. *arXiv preprint arXiv:2306.01245*.

CLaC at SemEval-2024 Task 4: Decoding Persuasion in Memes – An Ensemble of Language Models with Paraphrase Augmentation

Kota Shamanth Ramanath Nayak and **Leila Kosseim**
Computational Linguistics at Concordia (CLaC) Laboratory
Department of Computer Science and Software Engineering
Concordia University, Montréal, Québec, Canada
kotashamanthramanath.nayak@mail.concordia.ca,
leila.kosseim@concordia.ca

Abstract

This paper describes our approach to SemEval-2024 Task 4 subtask 1, focusing on hierarchical multi-label detection of persuasion techniques in meme texts. Our approach was based on fine-tuning individual language models (BERT, XLM-RoBERTa, and mBERT) and leveraging a mean-based ensemble model. Additional strategies included dataset augmentation through the TC dataset and paraphrase generation as well as the fine-tuning of individual classification thresholds for each class. During testing, our system outperformed the baseline in all languages except for Arabic, where no significant improvement was reached. Analysis of the results seem to indicate that our dataset augmentation strategy and per-class threshold fine-tuning may have introduced noise and exacerbated the dataset imbalance.

1 Introduction

The SemEval-2024 shared Task 4 (Dimitrov et al., 2024) proposed three distinct subtasks dedicated to identifying persuasion techniques conveyed by memes. The primary aim was to unravel how memes, integral to disinformation campaigns, employ various techniques to shape user perspectives. Subtask 1 focused on the analysis of textual content alone; while subtasks 2 and 3 involved the analysis of multimodal context that considers both textual and visual elements. Subtasks 1 and 2 used hierarchical multi-label classification metrics, while subtask 3 involves a binary classification task. The training dataset provided was in English but all subtasks mandated the evaluation of our model's zero-shot performance in three surprise languages: Bulgarian, North Macedonian, and Arabic and another fourth dataset in English. The goal during the testing phase was to explore our model's ability to generalize to these languages without explicit training.

This paper describes our participation to subtask 1, focusing on the detection of 20 persuasion techniques structured hierarchically within the textual content of memes. Inspired by successful approaches in multilabel text classification (Jurkiewicz et al., 2020; Tian et al., 2021), our strategy involved fine-tuning three language models i.e, BERT [`bert-base-uncased`], XLM-RoBERTa [`xlm-roberta-base`], and mBERT [`bert-base-multilingual-uncased`], followed by ensemble modeling using the mean aggregation technique using the English training set. To enhance performance, we used data augmentation through paraphrasing and adjusted the classification thresholds for each persuasion technique based on class-wise metrics optimised using the validation set using grid search. During testing, a zero-shot approach was implemented by translating the surprise language data into English.

At the shared task, our system demonstrated significant performance advantages over the baseline in all languages except Arabic, where the performance difference was not statistically significant. Our system's effectiveness, particularly in non-Arabic languages, underscores its potential for analyzing memes within disinformation campaigns, emphasizing the need for language-specific considerations in model development.

Section 2 provides an overview of the data utilized and offers insights into relevant prior research. Section 3 presents an overview of our classification pipeline, while Section 4 describes the experiments and data augmentation techniques that guided our final model decisions. Finally, Section 5 analyses the results of our model. All of the code used in the implementation of the models described in this paper is made available on GitHub.[1]

[1] https://github.com/CLaC-Lab/SemEval-2024-Task-4

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 175–180

2 Background

SemEval 2024 Task 4 (Multilingual Detection Of Persuasion Techniques In Memes) proposed 3 subtasks, out of which we participated in the first one. The goal of subtask 1 was to categorize the textual content of memes into one or several persuasion techniques. An inventory of 20 techniques was provided (eg: *Smears*, *Loaded Language*, *Slogans*) and were structured hierarchically, rendering the task a hierarchical multi-label classification problem.

2.1 Datasets

The SemEval organizers collected memes in English, Bulgarian, North Macedonian, and Arabic from their personal Facebook accounts, scraping public groups discussing politics, vaccines, COVID-19, gender equality, and the Russo-Ukrainian War. For subtask 1, the input data comprised the text extracted from these memes. The training (7k samples), validation (500 samples) and development (1k samples) sets included only English texts; whereas the test set was multilingual with 1500 samples for English, 426 samples for Bulgarian, 259 samples for North Macedonian and 100 samples for Arabic. All datasets were the provided in the form of JSON files. The orange bars in Figure 1 shows the distribution of the data for each persuasion technique in the training set. As Figure 1 shows some techniques, such as *Loaded Language* and *Smears*, had a substantial number of samples, while others like *Straw Man* and *Red Herring* were severely underrepresented.

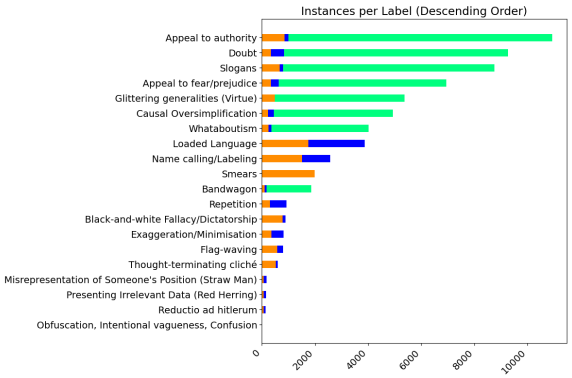


Figure 1: Distribution of the data for each persuasion technique in the SemEval 2024 (in orange), the `Comb-14k` (in orange + blue) and the `Para-54k` (in orange + blue + green) training datasets.

2.2 Previous Work

In the context of the SemEval 2020 Task 11 (Da San Martino et al., 2020), two subtasks were introduced addressing span identification of propagandistic textual fragments and a multi-label technique classification (TC) of propagandistic fragments using a corpus of ≈7k instances from the news domain. The subsequent SemEval 2021 Task 6 (Dimitrov et al., 2021) focused on the identification of propagandistic techniques from multimodal data including text and images from memes. This year's shared task build upon the 2021 task but included hierarchical metrics as well as a multilingual setting. The top-performing teams in 2020 and 2021, AppliCaAI (Jurkiewicz et al., 2020) and MinD (Tian et al., 2021) respectively, leveraged pre-trained language models and ensemble techniques to achieve top scores at the shared tasks. Inspired by these works, our methodology is also based on an ensemble of pre-trained language models.

3 System Overview

The aim of subtask 1 is to identify 0 or n persuasion techniques for each textual instance. Despite the hierarchical organization of the persuasion techniques, we opted to predicting solely the technique names (leaf nodes) and not their ancestor nodes. Figure 2 shows an overview of the classification pipeline we employed for this subtask. As shown in Figure 2, our methodology is based on fine-tuning three distinct pre-trained language models: BERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020), and mBERT (Devlin et al., 2019). This fine-tuning process is conducted on augmented datasets.

3.1 Data Augmentation

As Figure 1 shows, some persuasion techniques have very few samples (eg: *Red Herring*, *Straw Man* only have 59 and 62 instances respectively) in the SemEval 2024 dataset (in orange). To mitigate the lack of data we took advantage of data augmentation strategies: The Technique Classification subtask from SemEval 2020 task 11 (Da San Martino et al., 2020) (See Section 3.1.1) and automatically generated paraphrases (See Section 3.1.2).

3.1.1 SemEval 2020 Data (`Comb-14k` dataset)

The Technique Classification (TC) subtask from the SemEval 2020 Task 11 (Da San Martino et al.,

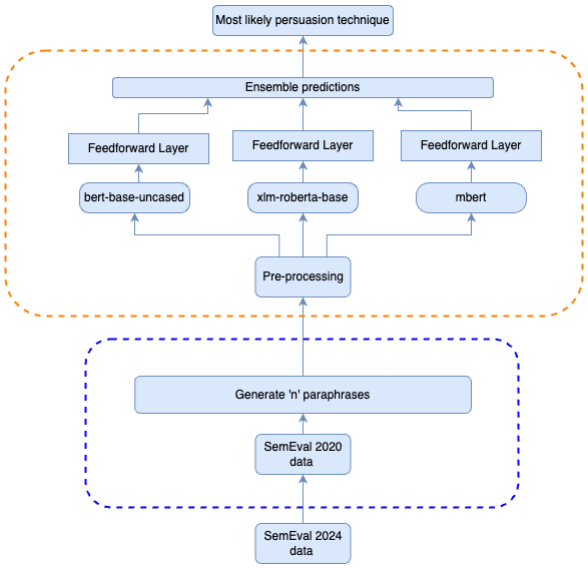


Figure 2: Schematic overview of our classification pipeline for the detection of persuasion techniques in memes.

2020) provided a dataset with ≈7k instances annotated with the same guidelines as this year's. In contrast to the 2020 task, this year's challenge featured a revised set of techniques compared to the 2020 inventory. In the 2020 TC dataset, a few techniques were merged into a single category due to lack of data, resulting in a list of 14 techniques. In the current year, an expanded inventory of 20 techniques was employed. To ensure consistency between the two sets, we preprocessed the 2020 TC dataset by splitting techniques that had previously been merged. For example, we singled out *Bandwagon* and *Reductio ad Hitlerum*, which had been merged into a single technique in the SemEval 2020 TC dataset.

We combined both datasets and fine-tuned models on this combined dataset. For easy reference in the rest of the paper, we call the combined dataset `Comb-14k`. Figure 1 (orange + blue) shows the resulting distribution of the persuasion techniques in this dataset.

3.1.2 Paraphrasing (`Para-28k`, `Para-52k` and `Para-54k` datasets)

Despite having almost doubled each class with the use of the 2020 TC dataset, some classes were still severely underrepresented; see Figure 1 (orange + blue). To address this, we augmented the dataset further by generating paraphrases for each instance. To generate paraphrases, we leveraged ChatGPT-3.5 turbo, setting the temperature to 0.7. This value aimed to introduce diversity in the paraphrases while maintaining relevance to the original instances.

For each instance in `Comb-14k`, we generated n paraphrases, then labeled these paraphrases with the same set of labels as the original instance. We experimented with n=1 and n=3. We call the resulting datasets `Para-28k` and `Para-52k`. The overall hierarchical F-score with the validation set given showed an increase when training with these datasets and $n = 3$ seemed to perform better than $n = 1$. A per-class analysis showed that not all classes benefited from the increase in support. For example, the persuasion technique *Bandwagon* increased its F1 from 0.17 to 0.29; whereas *Repetition* decreased its F1 from 0.56 to 0.31. We therefore identified the classes with improvement in F-score greater than 0.03 when using the `Para-52k` dataset compared to the `Comb-14k` dataset. These 8 techniques along with their increase in F-scores are shown in Table 1. This set of 8 techniques, referred to as benefited classes $\mathbf{B}$, formed the basis for our subsequent strategy. Since only these techniques seemed to benefit from the use of paraphrases, we only increased the number of paraphrases for these. Specifically, for all data instances d_i in `Comb-14k` labeled with techniques $\mathbf{T} = \{T_1, T_2, \ldots T_n\}$, for each $T_i \in \mathbf{B}$, we generate 10 paraphrases of d_i and label them with all techniques from $\mathbf{T} \cap \mathbf{B}$. This newly created dataset contained ≈54k instances, hence we call it `Para-54k`.

Figure 1 shows the distribution of instances for each technique in the `Para-54k` dataset (orange + blue + green), in comparison with the SemEval 2024 dataset and the `Comb-14k` dataset. As the figure shows, all datasets are severely imbalanced; something that we tried to address with the use of per-class custom thresholds (see Section 3.2).

3.2 Multi-label Classification

After creating the datasets, we preprocessed them using standard tokenization, then proceeded to fine-tune three distinct models: `bert-base-uncased`, `xlm-roberta-base`, and `bert-base-multilingual-uncased` in addition to an ensemble model, generated by averaging the predictions from all three models.

Additionally, we implemented thresholding in order to determine which techniques have a high enough score to be part of the output label set. We experimented with custom values for each of the techniques in order to address the data imbalance

Technique	Comb-14k		Para-52k		
	Support	F1	Support	F1	Δ F1
Bandwagon	169	0.17	676	0.29	0.12
Causal Oversimplification	449	0.00	1796	0.09	0.09
Appeal to fear/prejudice	631	0.26	2524	0.34	0.08
Doubt	843	0.08	3372	0.15	0.07
Appeal to authority	994	0.69	3976	0.74	0.05
Glittering generalities (Virtue)	488	0.38	1952	0.43	0.05
Slogans	796	0.42	3184	0.46	0.04
Whataboutism	366	0.32	1464	0.36	0.04

Table 1: Techniques that showed an improvement in F1 score when using n=3 paraphrases (i.e. Para-52k).

issue. We experimented with values ranging from 0.01 to 0.7 and picked the optimal values for each class based on the validation set (500 samples). These thresholds were applied to the scores obtained after passing the logits of each class through a sigmoid function. Table 2 shows the results of the validation with the optimal threshold for each class using the official scorer, which uses hierarchical metrics. As Table 2 shows, the best model with the validation set was the ensemble trained on the Para-52k dataset which reached an hierarchical F1 of 0.56. However, the ensemble model when trained on the Para-54k dataset, performed worse (hierarchical F1 of 0.54 with the validation set) than the ones that used lesser number of paraphrases (Para-28k and Para-52k). The ensemble, leveraging the collective insights of the three models, trained on the Para-52k emerged as the most effective in enhancing the overall system performance. Based on our results in the official leaderboard with the development set and validation results shown in Table 2, we chose to submit the ensemble model trained on the Para-52k dataset as it gave the best results with both the validation and the development set.

During the testing phase, the datasets in Bulgarian, North Macedonian, and Arabic were automatically translated to English for our model's zero-shot predictions. This was inspired by the approach of (Costa et al., 2023). The English test data was used as given.

4 Experimental Setup

4.1 Data Split and Augmentation

The training data provided in English initially comprised 7k samples. After combining it with 2020 TC dataset, the total increased to approximately 14k samples (Comb-14k). Subsequently, through paraphrase generation, the training dataset expanded to around 28k (Para-28k) when only 1 paraphrase per instance was used (n=1) and 52k (Para-52k), when n=3. Finally, the dataset with ten paraphrases for the benefited classes *B* reached approximately 54k samples (Para-54k). The original 500-sample validation set was used consistently for all our experiments. For the final submission, the ensemble model was trained on the union of (Para-52k) and the development set (1k samples), for a total of 53k samples.

4.2 System Pipeline and Training Details

The system pipeline code was implemented in PyTorch. The pre-trained models BERT [bert-base-uncased][2], XLM-RoBERTa [xlm-roberta-base][3], and mBERT [bert-base-multilingual-uncased][4] and their tokenizers were sourced from Hugging Face. Standard preprocessing, involving tokenization based on each model's tokenizer, was applied. Across all phases, models were trained for 10 epochs using the Adam optimizer with a learning rate of 2e-5. Batch sizes varied with BERT utilizing 128, and XLM-RoBERTa and mBERT using 64. A final feedforward layer with 20 logits (equal to the number of considered techniques) was added to each model. The Binary Cross Entropy with logits served as the loss function, with one-hot encoding applied to the true labels. For prediction, a sigmoid activation function was used on the logits, followed by thresholding. The ensemble model used an unweighted average of all predictions from the three individual models.

[2] huggingface.co/bert-base-uncased
[3] huggingface.co/FacebookAI/xlm-roberta-base
[4] huggingface.co/bert-base-multilingual-uncased

Training Set Used	Models	Validation Set	Development Set
Comb-14k	BERT	0.52	0.55
	XLM-RoBERTa	0.53	0.54
	mBERT	0.53	0.54
	Ensemble Model	0.53	0.56
Para-28k	BERT	0.55	0.57
	XLM-RoBERTa	0.57	0.54
	mBERT	0.50	0.53
	Ensemble Model	0.55	0.56
Para-52k	BERT	0.54	0.55
	XLM-RoBERTa	0.54	0.54
	mBERT	0.54	0.55
	Ensemble Model	**0.56**	**0.57**
Para-54k	BERT	0.48	0.51
	XLM-RoBERTa	0.54	0.55
	mBERT	0.51	0.53
	Ensemble Model	0.54	0.55

Table 2: Hierarchical F1 scores of our models, when trained on different English-language datasets for both the validation and development sets.

Language	Baseline	Our Score	Best Score
English	0.36865	0.57827	0.75427
Bulgarian	0.28377	0.44917	0.56833
North Macedonian	0.30692	0.39471	0.51244
Arabic	0.35897	0.38070	0.47593

Table 3: Comparison of the final hierarchical F1 scores obtained by our classification system, the best corresponding classification system in the shared task and the baseline in each given language.

ChatGPT-3.5 turbo[5] API with a temperature set to 0.7 was used for paraphrase generation. During testing, external languages were translated into English using the deep-translator API[6].
Throughout all phases hierarchical metrics were employed for task evaluation using the official scorer. On the other hand, standard precision, recall, and F-score metrics were used to assess the per class performance.

5 Results

The official performance results of our system are shown in Table 3, along with the baseline score and the score obtained by the best performing system on each language. As Table 3 shows, although our ensemble model was not among the top models, it reached significantly better performance than the baseline in all languages except Arabic, where the improvement was not significant. Overall, we stood at 22nd out of 33 participants for English, 12th out of 20 for Bulgarian, 11th out of 20 for North Macedonian and 11th out of 17 for Arabic.

6 Conclusion

This paper described the methodology used in our participation to the Semeval 2024 Task 4 subtask 1, focusing on hierarchical multi-label detection of persuasion techniques in meme texts. We used an ensemble model with three fine-tuned language models and incorporated additional strategies such as data augmentation through paraphrasing and classification thresholds fine-tuning based on class-wise metrics. During testing, our system significantly outperformed the baseline in all languages except Arabic, where the increase in performance was not significant. Analysis shows that the data augmentation and threshold fine-tuning may have introduced noise and exacerbating dataset imbalance.

[5] https://platform.openai.com/docs/models/gpt-3-5-turbo
[6] https://pypi.org/project/deep-translator/

Acknowledgements

The authors would like to thank the organisers of the SemEval shared task and the anonymous reviewers for their comments on the previous version of this paper. This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Nelson Filipe Costa, Bryce Hamilton, and Leila Kosseim. 2023. CLaC at SemEval-2023 task 3: Language potluck RoBERTa detects online persuasion techniques in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1613–1618, Toronto, Canada. Association for Computational Linguistics.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. ApplicaAI at SemEval-2020 task 11: On RoBERTa-CRF, span CLS and whether self-training helps them. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1415–1424, Barcelona (online). International Committee for Computational Linguistics.

Junfeng Tian, Min Gui, Chenliang Li, Ming Yan, and Wenming Xiao. 2021. MinD at SemEval-2021 task 6: Propaganda detection using transfer learning and multimodal fusion. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1082–1087, Online. Association for Computational Linguistics.

RDproj at SemEval-2024 Task 4: An Ensemble Learning Approach for Multilingual Detection of Persuasion Techniques in Memes

Yuhang Zhu
Uppsala University / Uppsala, Sweden
yuhang.zhu.2485@student.uu.se

Abstract

This paper introduces our bagging-based ensemble learning approach for the SemEval-2024 Task 4 Subtask 1, focusing on multilingual persuasion detection within meme texts. This task aims to identify persuasion techniques employed within meme texts, which is a hierarchical multilabel classification task. The given text may apply multiple techniques, and persuasion techniques have a hierarchical structure. However, only a few prior persuasion detection systems have utilized the hierarchical structure of persuasion techniques. In that case, we designed a multilingual bagging-based ensemble approach, incorporating a soft voting ensemble strategy to effectively exploit persuasion techniques' hierarchical structure. Our methodology achieved the second position in Bulgarian and North Macedonian, fifth in Arabic, and eleventh in English.

1 Introduction

Memes have gained immense popularity among the younger generation due to their entertaining nature. However, some memes can lead teenagers towards extreme ideas by employing persuasion techniques. Even well-educated people often need help to identify misleading memes. Thus, the development of a persuasion detection system holds significant value. This study aims to create a system to identify persuasion techniques within meme texts. This task is a multilabel and hierarchical classification task since memes may contain multiple persuasion techniques, and techniques have hierarchical structure (Dimitrov et al., 2024).

A description of the corpus provided by SemEval-2024 Task 4 (Dimitrov et al., 2024) reveals significant imbalances in the training data for the techniques. For instance, while there are 1990 instances for the "Smears" technique, only 258 instances pertain to "Whataboutism." Moreover, the training data for each technique is smaller compared with the entire corpus, leading to the imbalance between positive and negative instances for each technique. These observations lead us to formulate the following research questions: 1) How can we mitigate the data imbalance between techniques? 2) How can we ease the imbalance between positive and negative instances for each technique? 3) How can we effectively leverage the hierarchical structure of techniques? We devise a bagging-based ensemble learning system employing a soft voting strategy to solve these questions. We group techniques into ten subsets based on the amount of their training data and the hierarchical structure (Dimitrov et al., 2024), and construct a training set for each subset. Subsequently, we train classifiers (base learners), XLM-RoBerta$_{large}$[1] models with a classifier head, on these training sets. Finally, we compute the final distribution through a weighted average of the probability generated by classifiers, with a model of identical structure generating the weights in this step.

While our approach attained the second position in Bulgarian and North Macedonian, fifth in Arabic, and eleventh in English, the performance of our weight model did not exhibit significant improvement compared to our baseline. Moreover, the lower-resource techniques continue to suffer from imbalances between positive and negative instances. Our code is publicly available at https://github.com/Yuhang-Zhu-nlp/semeval2024_RDproj.

2 Background

2.1 Persuasion Detection

Previous research on persuasion detection has explored traditional classification techniques across a range of domains. Regarding data augmentation, Modzelewski et al. (2023) experimented with

[1] https://huggingface.co/FacebookAI/xlm-roberta-large

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 181–187

enhancing performance by expanding the training set using the DeepL API to translate data from source languages to target languages. Similarly, Falk et al. (2023) introduced a data augmentation method based on back-translation in the same year. Regarding text representation, Qachfar and Verma (2023) proposed a technique to generate language-agnostic features specific to this task, which were then concatenated with the CLS representation provided by XLM-RoBERTa to generate the final representation. Ensemble learning has also been explored in this domain. Purificato and Navigli (2023) developed a multilingual bagging-based ensemble learning system, combining five different BERT models using a soft voting strategy. Because of BERT's exceptional performance in sentence classification tasks, it has become a cornerstone in recent research, with almost all contemporary studies incorporating BERT into their methodologies (Costa et al., 2023; Ojo et al., 2023).

2.2 Ensemble Learning

The term ensemble learning is basically to improve the model's performance by combining different models (base learners) (Dong et al., 2020). Presently, ensemble learning strategies primarily include bagging, boosting, and stacking. Among these, bagging is training models on distinct datasets and combining them. One of the most renowned bagging-based ensemble learning algorithms is random forest (Cutler et al., 2012), which trains numerous decision trees on different data subsets and then combines these trees using a voting strategy. Regarding voting strategies, there are two main approaches: hard voting (Mohamed Kamr and Mohamed, 2022) and soft voting (Purificato and Navigli, 2023). Soft voting generates the final distribution by computing the weighted average of distributions from base learners, and has become a prevalent strategy in classification tasks (Xu et al., 2016; Kumari et al., 2021). Purificato and Navigli (2023) devised a bagging-based multilingual ensemble learning approach, employing five different BERT models with a soft voting strategy in this task. Their approach secured the first position in English during SemEval 2023, underscoring the effectiveness of bagging-based ensemble learning in this context. However, their approach determined model weights based on the normalized F1-micro score of diverse BERT models, ignoring the potential variability in model performance across different techniques.

2.3 Data

We use both the corpus offered by SemEval-2024 Task 4 Subtask 1 (Dimitrov et al., 2024) which contains English text of memes with 20 persuasion techniques and the corpus provided by SemEval-2023 Task 3 Subtask 3 (Piskorski et al., 2023) which includes news articles in six languages, English, German, French, Russian, Polish, and Italian, with 23 techniques.

3 System Overview

3.1 Data Preprocessing

In this task, we only focus on 20 techniques, but the corpus provided by SemEval-2023 Task 3 Subtask 3 contains 23 techniques. In that case, We have simply removed the three extra techniques from the label set of each data. The corpus provided by SemEval-2024 Task 4 Subtask 1 includes lots of meaningless symbols like "\n", we just simply remove them from the text. Moreover, we lowercase all data of both corpora.

3.2 Technique Grouping

To utilize the hierarchical structure of techniques, we categorize them into seven subsets based on their hierarchical structure (Dimitrov et al., 2024). For each subset, we assess whether data imbalance exists among the techniques. If imbalances exist, we create new subsets and copy the affected techniques or divide the subset into smaller subsets. For example, in the initial grouping, "Loaded Language", "Exaggeration/Minimisation", "Flag-waving", and "Appeal to fear/prejudice" are grouped in a subset. However, the training data for "Loaded Language" significantly outnumbers those for the other three techniques, so we separate "Loaded Language" into a new subset while removing it from the original subset. Additionally, if some techniques unavoidably suffer from data imbalances, we copy them to a new subset (supporting subset). Through this process, we ultimately establish ten distinct subsets, and the results of grouping are shown in Appendix.

3.3 Corpus Creating

For each technique subset, we first sample all data in the corpus provided by Semeval-2023 Task 3 Subtask 3 (Piskorski et al., 2023) (in the following section, we call it positive data). Then, we sample the data without techniques in the subset (in the following section, we call it negative data). Next,

we create the second corpus by doing the above step in the corpus offered by Semeval-2024 Task 4 Subtask 1 (Dimitrov et al., 2024).

3.4 Model Structure

We have 11 models in our approach, including 10 base learners and a weight model. All models have the same structure which is shown in Figure 1.

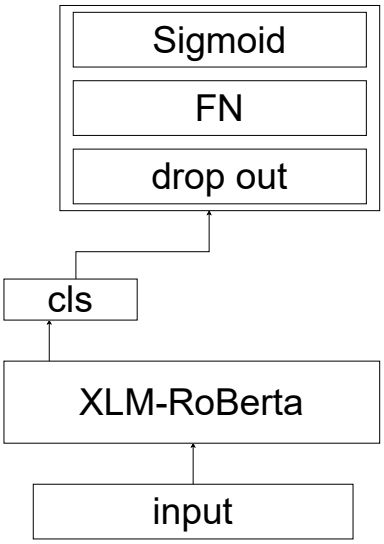


Figure 1: The structure of the base learners, and the weight model.

3.5 Training Strategy

Firstly, for each corpus sampled in the corpus provided by Semeval-2023 Task 3 Subtask 3 (Piskorski et al., 2023), we train a base learner on it (we call it pretrain in the following text). Then we fine-tune a base learner on each corpus sampled in the corpus offered by Semeval-2024 Task 4 Subtask 1 (Dimitrov et al., 2024). The task for base learners is to predict which persuasion techniques are applied in the given text. As for the weight model, we pretrain it on the original corpus provided by Semeval-2023 Task 3 Subtask 3 (Piskorski et al., 2023), and then fine-tune it on the corpus offered by Semeval-2024 Task 4 Subtask 1 (Dimitrov et al., 2024). The task of the weight model is to predict which technique subsets the persuasion techniques used in the given text belong to.

3.6 Prediction Pipeline

The prediction pipeline begins with preprocessing the text, which involves lowercasing and removing meaningless symbols. Subsequently, the text is sent to each base learner to obtain the technique distributions from each base learner. Similarly, the text is also sent to the weight model, and the output of the weight model is activated using softmax to generate the weight for soft voting. Finally, the final distribution is calculated using Equation (1).

$$D_{final} = \sum_{i=0}^{10} w_i D_i \tag{1}$$

where D_{final} is the final distribution, D_i is the distribution generated by the i^{th} base learner, and w_i is the weight generated by weight model for the i^{th} base learner.

4 Experimental Setup

We use binary cross-entropy (BCE) with weight as our loss function for each base learner. The equation is below:

$$\mathcal{L}(x_j, y_j) = \sum_{j=0}^{20} w_j (y_j log x_j - (1-y_j) log(1-x_j)) \tag{2}$$

where w_j is the weight for the j^{th} technique, y_j is the boolean value for the j^{th} technique, and x_j is the probability generated by the model for the j^{th} technique. We use BCE without weight for the weight model.

4.1 Training Setup

Each base learner has three hyperparameters: weights in the loss function, learning rate, and dropout rate. We set the learning rate to 2e-6 and the dropout rate to 0.2 for all base learners. The weights assigned to techniques belonging to the subset used to create the corpus on which the base learner is trained are set to 2, while all other techniques are assigned a weight of 1. Similarly, we use the same learning and dropout rates for the weight model as the base learner. During pretraining, we train each base learner for 60 epochs and the weight model for 50 epochs. During fine-tuning, we train each base learner for 20 epochs and the weight model for 10 epochs. The batch size is set to 16 for base learners and 8 for the weight model. we select 0.22 as our classification threshold.

4.2 Evaluation Metrics

Hierarchical-F1 (Kiritchenko et al., 2006) is used in this research. The benefit of the hierarchical-F1 is that it takes the hierarchical structure of techniques into account.

5 Results

5.1 Official Ranking

Table 1 shows our results in SemEval-2024 Task 4 Subtask 1. Although we get only the eleventh position in English, our results in three languages that are used to test zero-shot are competitive. We achieve the second position in both Bulgarian and North Macedonian, and the fifth position in Arabic.

5.2 Weight Model

We design a baseline model by removing the weight model, and set the weights for soft voting as $\frac{1}{10}$. In Table 2, we can find that our baseline and approach get almost the same score in English, Bulgarian, and North Macedonian. However, our baseline gets a relatively higher score in Arabic, which means that our weight model does not work well.

5.3 Error Analysis

In this section, we are aiming to find out the behaviour of our model facing different inputs by analyzing the samples which make our model give a wrong prediction in the dev set provided by SemEval-2024 Task 4 Subtask 1.

> **Text**: IF YOU SAY WE'RE IN THE MIDDLE OF A DEADLY PANDEMIC BUT YOU STILL SUPPORT OPEN BORDERS\\n\\nYOU'RE EITHER A LIAR OR A COMPLETE MORON
>
> ---
>
> **Gold labels**: Loaded Language, Name calling/Labeling, Black-and-white Fallacy/Dictatorship, Smears
>
> ---
>
> **Our prediction**: Appeal to fear/prejudice, Black-and-white Fallacy/Dictatorship, Loaded Language, Name calling/Labeling, Smears
>
> ---
>
> **Weight vector**: 0.0748, 0.0748, 0.0748, 0.1978, 0.2029, 0.0749, 0.0752, 0.0748, 0.0750, 0.0748

In this sample, we correctly identify all gold labels but detect "Appeal to fear/prejudice" by mistake. Analysis of the weight vector reveals that our weight model assigns a relatively higher weight of 0.2029 to the base learner trained on the corpora sampled for the subset (we call the base learner trained on the subset in the following text) containing "Appeal to fear/prejudice". However, it does not assign higher weights to subsets containing other techniques in the gold labels, except for "Loaded Language". To comprehend why our model can still make correct predictions despite the weight model's failure, we examine the output of several base learners. We observe that almost all base learners assign high probabilities to "Loaded Language", "Name calling/Labeling", and "Smears", indicating that each base learner can support techniques not included in the subsets on which they are trained. This suggests that each base learner can support the target techniques that are not included in the subsets they trained on.

> **Text**: Name: Ted Bundy\\nVictims: 30\\n\\nName: Al Gore\\nVictims: ???
>
> ---
>
> **Gold labels**: Reductio ad hitlerum, Smears
>
> ---
>
> **Our prediction**: Name calling/Labeling
>
> ---
>
> **Weight vector**: 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000

In this sample, we can find that our weight model does not work and give every subset a same weight. "Reductio ad hitlerum" is included in three technique subsets, and only the base learner trained on the supporting subset gives a high probability for this technique. However, other base learners give a very low probability, which shows our idea to create more subsets to support techniques suffering from data imbalance is working. The reasons for why we cannot distinguish "Reductio ad hitlerum" are 1) weight model cannot find which subsets the final prediction should be in, 2) positive and negative instances for "Reductio ad hitlerum" are too imbalanced, and our model tends to give a low probability.

> **Weight vector**: IS THE BUNDY SHOOTOUT A FALSE FLAG?\\n
>
> ---
>
> **Gold labels**: Doubt
>
> ---
>
> **Our prediction**: Loaded Language, Name calling/Labeling, Doubt
>
> ---
>
> **Weight vector**: 0.1663, 0.0958, 0.0922,

language	rank/nt	F1	T1F1
English	11/34	0.64288	0.75247
Bulgarian*	2/20	0.54089	0.56833
North Macedonian*	2/20	0.49869	0.51244
Arabic*	5/17	0.41129	0.47593

Table 1: The ranking of our approach in the official ranking of SemEval-2024 Task 4 Subtask 1. Languages with star are to test zero-shot. nt is the number of teams. F1 is the hierarchical-F1 score. T1F1 is the hierarchical-F1 score of the top-1 approach.

language	Our Model	Baseline
English	0.64288	0.64194
Bulgarian*	0.54089	0.54133
North Macedonian*	0.49869	0.49894
Arabic*	0.41129	0.41454

Table 2: The hierarchical-F1 score of our approach and the baseline on the test set.

0.0922, 0.0923, 0.0922, 0.0922, 0.0922, 0.0922, 0.0922

The weight model gives a higher weight for the first two subsets, which is correct because both subsets contain "Doubt". Almost all base learners give a high probability for "Doubt", which provide another evidence that base learners trained on other subsets can support gold labels. However, some base learners also give high probabilities for other two techniques in our prediction, resulting in wrong prediction. We should find a way to expand the gap between the weight of base learners trained on the subsets that include gold labels and on other subsets.

We can find some common elements in all samples. For example, "Loaded Language" and "Name calling/Labeling" are always predicted by mistake. A possible reason for this is that 0.22 is a reasonable threshold for some techniques but too small for some techniques which have rich training instances. Moreover, the accuracy of the weight model is not high enough.

6 Conclusion

In this study we build a persuasion detection system to distinguish which techniques are used in the given text of memes. Our system consists of ten base learners trained on different technique subsets and a weight model to generate the weight for soft voting. In the official ranking of SemEval-2024 Task 4 Subtask 1, we get competitive results in the zero-shot setting. However, our weight model does not work very well, and does not show a significant improvement compared with our baseline. The problems may be 1) the accuracy of the weight model is not high enough, 2) the gap between the weight of base learners trained on target subsets and other base learners is not big enough. Our idea to create a new technique subset to support techniques suffering from data imbalance seems feasible but the data imbalance between positive and negative instances of a technique is still a problem. The above discussion suggests the ideas to improve our approach. Firstly, we can improve the accuracy of the weight model by applying some new training techniques because our training method is very simple. Secondly, we need a more sophisticated technique grouping strategy which considers imbalance of positive and negative instances of a technique better.

Acknowledgements

I would like to express my appreciate for all participants who published their work in previous SemEval conferences. Additionally, I want to thank my teacher, Joakim Nivre, who gave me useful advice for my research.

References

Nelson Filipe Costa, Bryce Hamilton, and Leila Kosseim. 2023. CLaC at SemEval-2023 task 3: Language potluck RoBERTa detects online persuasion techniques in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1613–1618, Toronto, Canada. Association for Computational Linguistics.

Adele Cutler, D Richard Cutler, and John R Stevens. 2012. *Random forests*, pages 157–175. Springer.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International*

Workshop on Semantic Evaluation, SemEval 2024, Mexico City, Mexico.

Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258.

Neele Falk, Annerose Eichel, and Prisca Piccirilli. 2023. NAP at SemEval-2023 task 3: Is less really more? (back-)translation as data augmentation strategies for detecting persuasion techniques. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1433–1446, Toronto, Canada. Association for Computational Linguistics.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*, pages 395–406. Springer.

Saloni Kumari, Deepika Kumar, and Mamta Mittal. 2021. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. *International Journal of Cognitive Computing in Engineering*, 2:40–46.

Arkadiusz Modzelewski, Witold Sosnowski, Magdalena Wilczynska, and Adam Wierzbicki. 2023. DSHacker at SemEval-2023 task 3: Genres and persuasion techniques detection with multilingual data augmentation through machine translation and text generation. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1582–1591, Toronto, Canada. Association for Computational Linguistics.

Abdulrahman Mohamed Kamr and Ensaf Mohamed. 2022. akaBERT at SemEval-2022 task 6: An ensemble transformer-based model for Arabic sarcasm detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 885–890, Seattle, United States. Association for Computational Linguistics.

Olumide Ojo, Olaronke Adebanji, Hiram Calvo, Damian Dieke, Olumuyiwa Ojo, Seye Akinsanya, Tolulope Abiola, and Anna Feldman. 2023. Legend at ArAIEval shared task: Persuasion technique detection using a language-agnostic text representation model. In *Proceedings of ArabicNLP 2023*, pages 594–599, Singapore (Hybrid). Association for Computational Linguistics.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Antonio Purificato and Roberto Navigli. 2023. APatt at SemEval-2023 task 3: The sapienza NLP system for ensemble-based multilingual propaganda detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 382–388, Toronto, Canada. Association for Computational Linguistics.

Fatima Zahra Qachfar and Rakesh Verma. 2023. ReDASPersuasion at SemEval-2023 task 3: Persuasion detection using multilingual transformers and language agnostic features. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2124–2132, Toronto, Canada. Association for Computational Linguistics.

Steven Xu, HuiZhi Liang, and Timothy Baldwin. 2016. UNIMELB at SemEval-2016 tasks 4A and 4B: An ensemble of neural networks and a Word2Vec based model for sentiment classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 183–189, San Diego, California. Association for Computational Linguistics.

A Grouping of Technique Labels

subset	techniques
Ethos_ad	Name calling/Labeling
	Doubt
	Smears
	Reductio ad hitlerum
	Whataboutism
Ethos_ad_s	Doubt
	Reductio ad hitlerum
	Whataboutism
Ethos_ot	Bandwagon
	Appeal to authority
	Glittering generalities (Virtue)
Pathos_m1	Loaded Language
Pathos_m2	Exaggeration/Minimisation
	Flag-waving
	Appeal to fear/prejudice
Logos_JU	Bandwagon
	Appeal to authority
	Flag-waving
	Appeal to fear/prejudice
	Slogans
Logos_ot	Repetition
	Obfuscation, Intentional vagueness, Confusion
Logos_DI	Whataboutism
	Misrepresentation of Someoneś Position (Straw Man)
	Presenting Irrelevant Data (Red Herring)
Logos_SI	Causal Oversimplification
	Black-and-white Fallacy/Dictatorship
	Thought-terminating cliché
support_imbalance	Bandwagon
	Reductio ad hitlerum
	Obfuscation, Intentional vagueness, Confusion

Table 3: Grouping of Technique Labels

HausaNLP at SemEval-2024 Task 1: Textual Relatedness Analysis for Semantic Representation of Sentences

Saheed Abdullahi Salahudeen[1,2], **Falalu Ibrahim Lawan**[2], **Aliyu Yusuf**[3], **Amina Abubakar Imam**[4], **Lukman Aliyu, Nur Bala Rabiu**[5], **Mahmoud Said Ahmad, Idi Mohammed**[6], **Aliyu Rabiu Shuaibu**[7], **Alamin Musa, Auwal Shehu Ali**[8], **Zedong Nie**[1]

[1]Shenzhen Institute of Advanced Technology, CAS, [2]Kaduna State University, [3]Universiti Teknologi PETRONAS, [4]University of Abuja, [5]Khalifa Isyaka Rabiu University, [6]AUST, Abuja [7]Nile University, [8]Bayero University Kano, [∀]HausaNLP.

Contact: zd.nie@siat.ac.cn

Abstract

Semantic Text Relatedness (STR), a measure of meaning similarity between text elements, has become a key focus in the field of Natural Language Processing (NLP). We describe SemEval-2024 task 1 on Semantic Textual Relatedness featuring three tracks: supervised learning, unsupervised learning and cross-lingual learning across African and Asian languages including Afrikaans, Algerian Arabic, Amharic, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu. Our goal is to analyse the semantic representation of sentences textual relatedness trained on mBert, all-MiniLM-L6-v2 and Bert-Based-uncased. The effectiveness of these models is evaluated using the Spearman Correlation metric, which assesses the strength of the relationship between paired data. The finding reveals the viability of transformer models in multilingual STR tasks.

1 Introduction

The rapid increase in digital information has presented a critical challenge for researchers. The web hosts around 50 million pages of text, which is beyond the capacity of human interpretation alone. To interpret this extensive text data effectively, it is essential to comprehend the meanings of various words (Jain et al., 2020). Semantic Text Relatedness (STR) is a semantic analysis of the relationship between two pieces of text based on their meanings. STR of two language units has long been considered fundamental to understanding meaning (Miller and Charles, 1991; Lastra-Díaz and García-Serrano, 2015), It's a metric used to measure the similarity in meaning between two terms or documents. It is a subset of computational linguistics and one of the fundamental concepts of Natural Language Processing (NLP). STR can be measured using datasets designed by experts, which are made up of word pairs that are known to be related. It can be used in identifying a paraphrase or duplicate, as well as search engines to give users relevant and personalized results.

When two sentences have a paraphrase or entailment relation, they are considered to be semantically similar and When evaluating the semantic relatedness between them, humans typically focus on identifying shared meanings. In the case of the sentence pairs below, most English speakers would agree that the sentences in the first pair are more closely related in meaning than those in the second pair, whether they are from the same topic, express the same view or originate from the same time period etc.(Abdalla et al., 2023).

Pair 1: **a.** *There was a lemon tree next to the house.*
b. *The boy enjoyed reading under the lemon tree.*

Pair 2: **a.** *There was a lemon tree next to the house.*
b. *The boy was an excellent football player.*

Previous NLP research has mainly dealt with semantic relatedness primarily in English language. However, in this task, we address a variety of languages, including Afrikaans, Algerian Arabic, Amharic, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu. The task featured the following tracks: Track A which is a supervised learning, track B is an unsupervised learning and Track C is a cross-lingual learning.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 188–192

2 Related Works

Sentences are considered semantically related when they share commonalities in meaning, such as paraphrasal or entailment relations. A study by (Abdalla et al., 2023) developed a Semantic Textual Relatedness dataset (STR-2022) to manually annotate English sentence pairs and explore the factors that contribute to the semantic relatedness of sentences. The dataset has been used to study the degree of semantic relatedness and the reliability of human intuition in determining the relatedness of sentence pairs while (Hasan et al., 2020) assessed the methods for semantic relatedness between words based on knowledge sources. These methods exploit features from both structural and statistical approaches, emphasizing on semantic representation, measures of semantic similarity, and knowledge-based text mining.
(Lastra-Díaz and García-Serrano, 2015) proposed Explicit Semantic Analysis (ESA), a recently introduced approach that signifies the meaning of texts by computing the semantic relevance of natural language texts. This approach assumes the need for substantial amounts of common sense and domain-specific knowledge, utilizing machine learning techniques to explicitly depict the meaning of any text. This is achieved by creating a weighted vector based on concepts from Wikipedia. ESA undergoes continuous development, ensuring a consistent expansion of its breadth and depth over time.

3 Task Description

STR Shared Task 1 (Ousidhoum et al., 2024b) consists of predicting the semantic relatedness of sentence pairs. Sentence pairs will be rank based on their closeness in meaning in 14 different languages. All sentence pairs will have manually determined relatedness scores between 0 (completely unrelated) and 1 (maximally related). Participants are provided with a gold label scores with a comparative annotation approach that led to a high reliability of the final relatedness rankings. The shared task consists of three tracks: supervised learning, unsupervised learning and cross-lingual learning. In this paper, we concentrate on all the three tracks.

3.1 Track A: Supervised

This track relies on labelled input and output training data. We used the labeled training datasets for 9 languages provided for the shared task which include: Algerian Arabic, Amharic, English, Hausa, Kinyarwanda, Marathi,Moroccan Arabic, Spanish and Telgu.

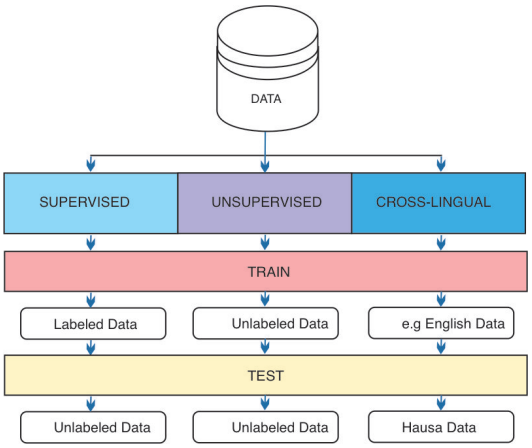


Figure 1: **Task Overview**

3.2 Track B: Unsupervised

Unsupervised learning analyzes and cluster unlabeled datasets, it is typically used when the goal is to identify patterns and relationships in data. We make this analysis using 12 languages: Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi,Indonesian, Kinyarwanda, Modern Standard Arabic, Moroccan Arabic, Punjabi and Spanish.

3.3 Track C: Cross-lingual

Cross-lingual learning involves transferring models from one language to another, typically to improve performance. For this track we make use of 12 languages: Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi,Indonesian, Kinyarwanda, Modern Standard Arabic, Moroccan Arabic, Punjabi and Spanish.

4 Experiment and Evaluation

This section describes the system overview which comprises the dataset description, model description and evaluation metric.

4.1 Dataset Description

The dataset consists of an instance of a sentence pair of both the training, development and test sets. Each instance is annotated with a gold label score that represents the degree of semantic text relatedness between two sentences (Ousidhoum et al., 2024a). The gold label scores are determine by manual annotation and range from 0 (not related

at all) to 1 (very related at all). A comparative annotation approach is used to avoid biases of the traditional rating scales and can result to a high reliability of final relatedness rankings. The dataset used in this shared task are from the following languages: Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Morrocan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu.

4.2 Models Description

We experiment with multiple pre-trained models before deciding to go with the selected models based on the tracks. However, due to time constraint and resources, we reported for the competitive models across various languages based on the task specification.

4.2.1 mBERT

We used mBERT in a supervised aapproach, mBert is a multilingual derivative of BERT and trained on a diverse set of 104 languages. The pre-training process for mBERT involves masked language modeling (MLM) and the next-sentence prediction task (Libovickỳ et al., 2019). To tailor the model for our specific task, we fine-tune the mBERT-base-cased model, which boasts 172 million parameters. A 70-30 train-test split is executed with a learning rate of 1e-5 on Adam optimizer.

4.2.2 all-MiniLM-L6-v2

The all-MiniLM-L6-v2 model was used in an unsupervised approach in this task, it is a lightweight transformer-based model for semantic similarity comparison with optimized model size and faster inference (Wang et al., 2020). It has 66 Million Parameters compressed in a Student-Mimicking-Teacher network relationship. Using self attention distribution, we utilized the Teacher's last layer to guide the training of the student distillation in an unsupervised manner and generated effective and flexible results for the 12 languages used.

4.2.3 BERT-BASED-UNCASED

The Bert-Based-Uncased model was used in a cross-lingual approach in this task. It is a pre-trained autoencoding language model trained on vast English Wikipedia and BookCorpus with a sequence length of 512. The model is based on the architecture presented in (Devlin et al., 2018). As the track description, some of the languages were initially trained on different language before applying task on new language. Bert-Based-Uncased use WordPiece tokenizer, it has 110 parameters 12-layer, 768-hidden, 12- attention heads.

Task	Model	Language	Sp. Corr.
Track A: Supervised	mBERT	Algerian Arabic	0.388
		Amharic	0.269
		English	0.762
		Hausa	0.580
		Kinyarwanda	0.527
		Marathi	0.811
		Moroccan Arabic	0.696
		Spanish	0.696
		Telugu	0.791
Track B: Unsupervised	all-MiniLM-L6-v2	Afrikaans	0.468
		Algerian Arabic	0.398
		Amharic	0.098
		English	0.825
		Hausa	0.273
		Hindi	0.465
		Indonesian	0.384
		Kinyarwanda	0.131
		Modern Standard Arabic	0.200
		Moroccan Arabic	0.496
		Punjabi	0.011
		Spanish	0.603
Track C: Cross-Lingual	BERT-BASED-UNCASED	Afrikaans	0.710
		Algerian Arabic	0.780
		Amharic	0.660
		English	0.780
		Hausa	0.630
		Hindi	0.740
		Indonesian	0.790
		Kinyarwanda	0.750
		Modern Standard Arabic	0.660
		Moroccan Arabic	0.670
		Punjabi	0.730
		Spanish	0.810

Table 1: Results of various tasks.

4.3 Spearman Correlation

The Spearman Correlation is a non parametric and normality for monotonic relationship between variables (Ali Abd Al-Hameed, 2022). It measures the strength of relationship between paired data. It is similar to Pearson's Product Moment Correlation Coefficient (De Winter et al., 2016), or Pearson's r. It indicates magnitude and direction of the association between two variables that are on interval

or ratio scale. For this task, we used Spearman Correlation to measure the similarity between two sentences.

5 Results and Discussion

This section presents the results of the Shared task Tracks. Table 1 displays the Spearman correlation scores for the evaluation of 14 low-resource languages for semantic relatedness. The SemEval-2024 task on STR provided an opportunity to explore the effectiveness of transformer models. The models capture semantic relatedness across multiple languages. In This section, the analyses and interpretations of the results obtained from the given tasks are Task A (supervised learning), Task B (unsupervised learning) and Task C (cross-lingual).

In supervised learning track A, the multilingual BERT (mBERT) model was used. The model demonstrated different levels of performance across the languages. Notably, mBERT exhibited strong correlation scores in languages such as English with 0.76, Marathi with 0.81, and Telugu with 0.79 correlation. This indicates the model's ability to generalize well across linguistic contexts in semantic relatedness tasks. These findings suggest that mBERT can effectively capture semantic relatedness, even in low-resource languages, highlighting its robustness and cross-lingual generalization capabilities. However, challenges were observed in languages with complex morphological structures, underscoring the need for further research to address such linguistic nuances.

Conversely, the unsupervised learning track B featured the All-MiniLM-L6-v2 model, which achieved promising results in certain languages, particularly English with 0.82, Spanish with 0.60, and Moroccan Arabic with 0.5 Spearman correlation value. Despite its effectiveness, the model faced difficulties in languages such as Punjabi and Amharic, where semantic relatedness was harder to capture without labelled data. These challenges emphasize the importance of developing techniques to improve unsupervised learning models' performance, especially in low-resource language settings.

Similarly, track C (cross-lingual) which were entirely trained with BERT-BASED-UNCASED performed promisingly despite training and predicting on different language pairs. The Spearman correlation for Spanish achieved 0.81 and was trained on English, while Hausa achieved the lowest with 0.63 despite being trained on Kinyarwanda training dataset. This performance especially in Semantic Textual Relationship shows that cross-lingual hold a prospective future for generalization of NLP tasks.

However, the findings highlight the effectiveness of transformer models, in capturing semantic relatedness across diverse languages. The choice of evaluation metrics, such as Spearman correlation, proved instrumental in assessing the models' performance and understanding their ability to capture the ordinal relationship between predicted and true semantic relatedness scores. Furthermore, the results contribute valuable insights into advancing the understanding and application of semantic textual relatedness in multilingual NLP tasks, paving the way for future research in this domain.

6 Conclusion and Future Works

The study on Semantic Text Relatedness (STR) across multiple languages has demonstrated the effectiveness of transformer models in capturing semantic relatedness. The multilingual BERT (mBERT) model showed strong correlation scores in languages such as English, Marathi, and Telugu, indicating its ability to generalize well across linguistic contexts. The All-MiniLM-L6-v2 model achieved promising results in English, Spanish, and Moroccan Arabic, while facing challenges in languages like Punjabi and Amharic. The cross-lingual track, using BERT-BASED-UNCASED, also performed well, especially in Spanish, trained on English data. These findings underscore the potential of transformer models in NLP tasks and the importance of appropriate evaluation metrics like Spearman Correlation. The study contributes valuable insights into advancing semantic textual relatedness in multilingual NLP, highlighting areas for future research and development.

Future work should focus on exploring advanced transformer Large Language Models (LLMs) like GPT-3 and T5 to improve performance across diverse languages, including low-resource ones. Expanding language coverage, incorporating contextual and cultural information, and fine-tuning with language-specific data will enhance model accuracy. Cross-lingual transfer learning techniques can be investigated to adapt high-resource language models to low-resource settings. Hybrid approaches combining different learning methods may offer improved results, while new evaluation

metrics could better capture semantic nuances. Additionally, exploring multimodal STR and applying research findings to real-world applications will increase the practical impact of STR systems.

Acknowledgements

We would like to acknowledge HausaNLP Management for their tireless support.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, Croatia. Association for Computational Linguisticst doi =.

Khawla Ali Abd Al-Hameed. 2022. Spearman's correlation coefficient in statistical analysis. *International Journal of Nonlinear Analysis and Applications*, 13(1):3249–3255.

Joost CF De Winter, Samuel D Gosling, and Jeff Potter. 2016. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods*, 21(3):273.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ali Muttaleb Hasan, Noorhuzaimi Mohd Noor, Taha H Rassem, and Ahmed Muttaleb Hasan. 2020. Knowledge-based semantic relatedness measure using semantic features. *International Journal*, 9(2).

Shivani Jain, KR Seeja, and Rajni Jindal. 2020. A new methodology for computing semantic relatedness: modified latent semantic analysis by fuzzy formal concept analysis. *Procedia Computer Science*, 167:1102–1109.

Juan J Lastra-Díaz and Ana García-Serrano. 2015. A novel family of ic-based similarity measures with a detailed experimental survey on wordnet. *Engineering Applications of Artificial Intelligence*, 46:140–153.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

A Appendix

Lang.	Sentence 1	Sentence 2	Score
Amharic	መግለጫውን የተከታተለው የአዲስ አበባው ዘጋቢያችን ሰሎሞን ሙጬ ዝርዝር ዘገባ አለው።	በስፍራው ተገኝቶ የተከታተለው የአዲስ አበባው ዘጋቢያችን ሰሎሞን ሙጬ ያጠናቅረውን ልከዋል	0.88
Moroccan Arabic	10 فالمغرب الصحية الطوارئ حالة تمديد	دجنب 10 فالمغرب الصحية الطوارئ حالة تمديد	0.72
Spanish	Una mujer a punto de comer trucha.	Una mujer a punto de comer pescado.	1.0
English	It that happens, just pull the plug.	if that ever happens, just pull the plug.	1.0
Hausa	Yan bindiga sun yi garkuwa da mutane 11 a Shimfidà, jihar Katsina	Yan bindiga sun yi garkuwa da dalibai mata a jihar Zamfara AN GUDU NA A TSIRA BA	0.59
Kinyarwanda	Ibicirizwa by'ababikiri bayo irabibungabunga Bimwe mu bikoresho Romeobuy	ibonera ababikiriya bayo Ijambo a muritegurirwa na Rejoice Ministries	0.19
Marathi	गुरु नानक देव आणि त्यांची शिकवण-दिक्षा संपूर्ण मानवजातीसाठी	केवळ भारतातीलच नाही, तर संपूर्ण मानवजातीसाठी मार्गदर्शक आहे	1.0
Algerian Arabic	ام فيها تاكل رانى تركا جريتها وليد ام الصحة يعطيك بنينة شحال	روعة جا مسمن ليوم جريت وليد ام الصحة يعطيك وفية مثلجمة زيان لو انا	0.62
Telugu	జమ్మూకాశ్మీర్లో మంచుకొండచరియలు విరిగిపడిన ఏలువ్వేరు	ఐదుగురు మృతి చెందారు	0.88
Afrikaans	My eerste stukkie advies is dat jy realisties moet wees oor die afstand wat jy wil hengel	Dit bring tot n einde die maanverkenningsprogram van die Verenigde State.	0.19
Indonesian	Pendidikan Desa Pusaka memiliki 4 sekolah.	Pendidikan Desa Serumpun Buluh memiliki 4 sekolah.	0.83
Hindi	{देश में कोरोना वायरस से मौत का आंकड़ा 100 के पार पहुंचा, पिछले 12 घंटे में 26 की गई जान}	{देश में कोरोना वायरस का कहर तेजी से बढ़ता जा रहा है।}	0.72
Punjabi	{ਪੰਜਾਬ ਤੇ ਦੂਜੀ ਵਾਰ ਦਿਹਾਇਕ ਬਣੇ ਅਮਨ ਅਰੋੜਾ ਦਾ ਮੰਤਰੀ ਬਣਨਾ ਤੈਅ ਹੈ}	{ਇਨ੍ਹਾਂ ਵਿੱਚ ਦੂਜੀ ਵਾਰ ਦਿਹਾਇਕ ਬਣੇ ਪ੍ਰੋ. ਬਲਜਿੰਦਰ ਕੌਰ ਜਾਂ ਸਰਦਜੀਤ ਮਾਣੂੰਕੇ ਨੂੰ ਵੀ ਮੰਤਰੀ ਬਣਾਇਆ ਜਾ ਸਕਦਾ ਹੈ}	0.56
Modern Standard Arabic	هذا الاعوجاج	هذه السحابة النفطية	0.83

Figure 2: Example Sentences

SCaLAR NITK at SemEval-2024 Task 5: Towards Unsupervised Question Answering system with Multi-level Summarization for Legal Text

M Manvith Prabhu* **Haricharana Srinivasa†** **Anand Kumar M§**
Department of Electronics and Communication *,
Department of Chemical Engineering †,
Department of Information Technology §,
National Institute of Technology Karnataka (NITK), Surathkal - 575025, India
{manvithprabhu.211ec228, sharicharana.211ch024, m_anandkumar}@nitk.edu.in

Abstract

This paper summarizes Team SCaLAR's work on SemEval-2024 Task 5: Legal Argument Reasoning in Civil Procedure. To address this Binary Classification task, which was daunting due to the complexity of the Legal Texts involved, we propose a simple yet novel similarity and distance-based unsupervised approach to generate labels. Further, we explore the Multi-level fusion of Legal-Bert embeddings using ensemble features, including CNN, GRU and LSTM. To address the lengthy nature of Legal explanation in the dataset, we introduce T5-based segment-wise summarization, which successfully retained crucial information, enhancing the model's performance. Our unsupervised system witnessed a 20-point increase in macro F1-score on the development set and a 10-point increase on the test set, which is promising given its uncomplicated architecture.

1 Introduction

The Domain of Law demands sheer expertise and experience for a human to master, but it takes much more to teach a machine the same. Legal NLP (Zhong et al., 2020) is advancing at a rapid pace, and the advent of Transformers (Vaswani et al., 2017) has widened the prospects of research in this area. However, the intricate nature of Legal Texts and the underlying complex relationships between entities make it difficult even for state-of-the-art Language models like BERT (Devlin et al., 2019) to capture the details effectively. To advance our understanding of the reasoning ability of LLMs in the legal domain (Bongard et al., 2022), task 5 of SemEval-2024 was proposed (Held and Habernal, 2024). The objective of this task is to discern the accurate responses to legal inquiries in U.S. Civil Procedure, as posited by the organizers. The questions and answers adhere to a Multiple-choice question-answering model, with accompanying explanations provided to facilitate comprehension of the legal concepts associated with each question. We have also released the code on GitHub [1]

We delve into the foundational paradigms of machine learning, specifically focusing on Supervised and Unsupervised Learning, to introduce innovative approaches and present a comprehensive comparative analysis. The explanation part of our dataset undergoes a two-level segment-wise summarization generated by T5 (Roberts et al., 2019), which is consistently utilized throughout our investigation. Within the framework of the supervised setup, we leverage a multi-level CNN fusion approach (Usama et al., 2019), integrating LSTM and GRU architectures. This amalgamation facilitates the extraction of ensemble feature representations from questions, answers, and summaries. Additionally, a one-dimensional CNN model (Jacovi et al., 2018), is trained. We employ a manual grid search technique to determine the optimal threshold that maximizes the macro F1 score, contributing to the refinement of our model.

In the unsupervised setup, we delve into the acquisition of diverse word representations such as word2vec and Glove. The assessment involves computing the similarity between question-answer pairs and answer-summary pairs, employing combinations like Glove-cosine, transformer embedding-cosine, transformer embedding-euclidean and word2vec-cosine. Notably, the best-performing supervised model achieved a macro F1 score of 66 % on the development set and 49.6 % on the test set. In contrast, the unsupervised approach yielded scores of 62 % (development) and 52.3 % (test). This outcome highlights a nuanced challenge related to generalization on the test set, prompting further exploration into the intricacies of model adaptability and robustness.

[1] https://github.com/haricharan189/Semeval_task5.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 193–199

2 Background

The dataset provided by the organizers comprises three sets: Train Set, Dev Set, and Test Set, containing 666, 84, and 98 data points, respectively. Within the training and dev sets, each entry includes fields such as Question, Answer, Explanation, Label (with values of 0 or 1), Analysis, and Complete-Analysis providing a detailed examination. The test set, on the other hand, only consists of Question, Answer, and Explanation. The Label, when equal to 1, signifies a correct answer, while 0 denotes an incorrect one. The Explanation field provides context and background details for each question.

Field	Text
Explanation	The most basic point to understand about supplemental jurisdiction ..... on this basic purpose of Âr-ticle 1367(a).
Question	This and that. Garabedian, ... are treated fairly.
Answer	has constitutional authority ..... under Ârticle 1367(a).
Label	0
Analysis	Here, the Ârticle 1983 claim ..... Amend-ment claim.
Complete analysis	This is pretty straightfor-ward ..... D is the best choice here.

Table 1: Sample data-point from Train Set.

3 Related Works

Legal texts pose a unique challenge for pre-trained transformers (Vaswani et al., 2017) due to the inclusion of specialized terminology not commonly used in everyday language . As a result, leveraging pre-trained models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and others becomes essential by training them on legal corpora to enhance their understanding of legal terminologies. Notable examples of transformers tailored for legal contexts include InLegalBERT (Paul et al., 2023), Legal-RoBERTa (Geng et al., 2021), and similar models.

Fine-tuning transformers, such as Legal-BERT (Chalkidis et al., 2020), on available legal data has been proposed as an effective strategy to improve performance on test sets, as suggested by Bongard et al. (2022) (Bongard et al., 2022). This approach capitalizes on domain-specific knowledge encoded during pre-training, enhancing the model's proficiency in handling legal language nuances.

In the domain of Legal Question Answering (LQA), recent works have extensively discussed significant advancements and challenges. The comprehensive review by Martinez-Gil provides insights into the key works in LQA, outlining challenges and proposing future research directions. Louis et al. (2023) (Louis et al., 2023) shed light on the limitations of existing Large Language Models (LLMs) in Legal Question Answering, emphasizing the need for interpretability.

4 System Overview

Transformers like T5, as demonstrated in the work of (Roberts et al., 2019), exhibit high efficiency in producing summaries for lengthy paragraphs. In this study, T5 was employed to generate segment-level summaries on explanation column using a two-step approach. The initial summary was created from the original text, with a segment length of 1000 tokens. These segment-wise summaries were then concatenated with spaces in between to form the first summary. Subsequently, the second summary was generated from the first summary, employing a segment length of 300 tokens, and similarly concatenated to provide a comprehensive summary of the input text. These summaries were used for further applications in place of explanation. Segment wise summary approach can be visualized as follows:

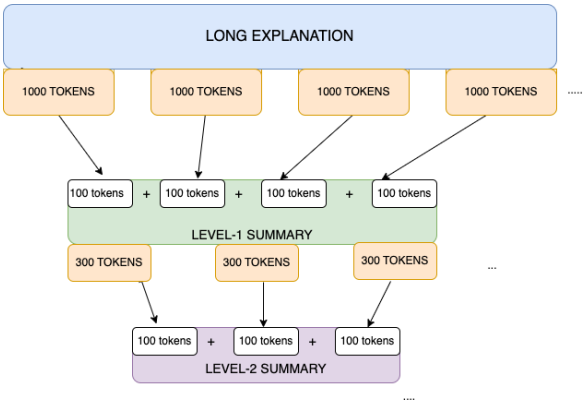


Figure 1: Segment wise summary

4.1 Supervised Models

4.1.1 Multi-Level Approach

Following the generation of summaries, we employed the Legal-Bert transformer to extract embeddings from the question, answer, and summary columns. Each Legal-Bert output consists of a 768-dimensional vector, resulting in tensors of shape (number of data points, 768) for each dataset. Subsequently, we executed the following steps:

1.The tensor underwent a series of transformations through three consecutive 1-dimensional CNN layers, with ReLU activation functions (Nair and Hinton, 2010), and Adaptive max-pooling applied at each step. At each pooling layer, the output was reduced to 100 dimensions. The kernel size and padding were linearly increased, as depicted in the Figure 2.

2.The outputs from the first and second pooling layers were concatenated, yielding a first-level concatenated feature embedding of 200 dimensions.

3.This first-level output was then merged with the output from the third pooling layer to obtain a second-level concatenated embedding with 300 features.

4. Concurrently, the Legal-Bert embeddings were fed into Bi-GRU (Chung et al., 2014) and Bi-LSTM (Hochreiter and Schmidhuber, 1997) models, resulting in 100 features from each. These features were concatenated.

5. The final multi-level feature representation was achieved by concatenating the second-level features with those from the GRU-LSTM models, resulting in a 500-dimensional vector. This process was applied to the question, answer, and summary, culminating in an exhaustive 1500-dimensional representation of the training data.

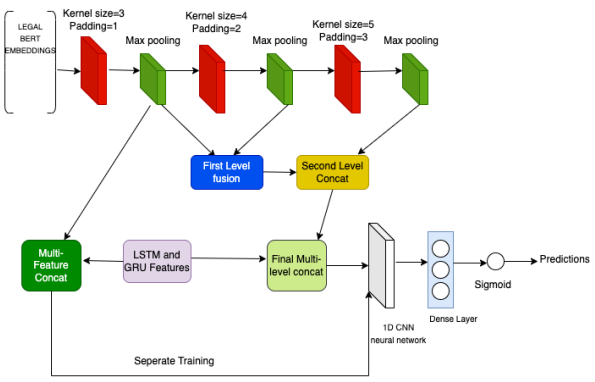


Figure 2: Multi Level fusion

4.1.2 Multi-Feature Approach

In this approach, the output of the first pooling layer was directly concatenated with the GRU-LSTM features, resulting in 300 features per entity, and hence, a 900-dimensional representation of the training data.

Training and custom sigmoid layer: To conduct a comparative analysis, we trained separate models using both multi-level and multi-feature representations. In each case, we employed a 1-dimensional CNN architecture implemented in TensorFlow, featuring a kernel size of 3 and 32 filters. Following max pooling, the resulting output was flattened and fed into a dense layer comprising 128 neurons. Finally, to enhance the variability of the probability distribution in the predictions, we introduced a custom Lambda layer. This layer subtracts the mean of the input tensor from each element and subsequently applies the sigmoid activation function.

$$\mathrm{f}(x) = y = \mathrm{sigmoid}(x - \mu) \qquad (1)$$

where μ is the mean of x

Grid search and predictions: Following the generation of probability vectors for the development set, we utilized manual grid search to determine the optimal threshold for classifying correct answers, aiming to maximize the macro-F1 score. Subsequently, the threshold associated with the highest F1 score on the development set was applied to make predictions on the test set

4.2 Unsupervised Models

4.2.1 Word2Vec-Cosine system

Word2Vec embeddings, as described in (Mikolov et al., 2013), were extracted for the question, answer, and summary columns. A window size of 7 and a vector size of 5 were utilized for each word. Cosine similarities were computed between question-answer pairs and answer-summary pairs. The prediction was based on the mean of these similarities.

During evaluation, it was observed that in cases where the difference between the highest and second-highest similarity scores for a question was minimal, the answer with the second-highest similarity often turned out to be the correct answer. Consequently, a refinement was implemented: if the disparity between the highest and second-highest similarity scores was small, the answer with the second-highest similarity was labeled as

1, while the remaining answers were labeled as 0. This adjustment yielded improved results in such scenarios. A threshold of 0.0005 was used in this case after optimization on train and dev set.

Algorithm 1: Word2Vec Similarity-based Labeling

```
Data: Word2Vec embeddings for question,
      answer, and summary columns
Result: Labels for answers based on
        similarity scores
for each question do
    max_id= highest similarity score;
    second_max_id = second-highest
     similarity score;
    if |similarity[max_id] −
     similarity[second_max_id]| ≤
     0.0005 then
        Label[second_max_id] = 1;
        Label the remaining answers as 0;
    end
    else
        Label[max_id] = 1;
        Label the remaining answers as 0;
    end
end
```

4.2.2 GloVE-Cosine system

In contrast to the Word2Vec-Cosine approach, the methodology now incorporates GloVE embeddings as opposed to Word2Vec embeddings, leveraging the GloVE model proposed by Pennington et al. in 2014 (Pennington et al., 2014). Despite this shift, the overarching algorithm for label assignment remains unaltered, ensuring continuity and comparability with the Word2Vec-Cosine approach discussed in the preceding section.

4.2.3 Transformer embeddings-Cosine system and Transformer embeddings-Euclidean system

We utilized the Deberta model (He et al., 2021) trained on legal texts, specifically "LambdaX-AI/legal-deberta-v1," accessible on Hugging Face (Wolf et al., 2020). This model provided embeddings of questions, answers, and summaries, each represented by vectors of size 1536. We employed both cosine similarity and Euclidean distance metrics for label assignment.

For cosine similarity, the algorithm remained straightforward: answers with higher cosine similarity scores were assigned labels accordingly.

However, in the case of Euclidean distance, a slightly different approach was employed. The answer with the minimum distance was initially assigned a label of 1. Subsequently, if the difference between the minimum distance and the second minimum distance was less than a predefined threshold which is 0.8 in this case, the answer associated with the second minimum distance was labeled 1 instead, replacing the initial assignment.

5 Experimental Setup

We utilized Google Colab for training and testing our models, taking advantage of the T4 GPU provided by the platform.

5.1 Supervised Models

The Multi-feature concatenation method involved the integration of 900 features, while the Multi-level approach incorporated 1500 features. Both methodologies underwent training for 15 epochs with a batch size of 32. The optimization algorithm chosen was "Adam" (Kingma and Ba, 2017), employing a learning rate set to 0.001.

5.2 Unsupervised Models

Word2Vec and GloVe embeddings were both generated with an embedding size of 5. However, there were differences in the window length used during training: for Word2Vec embeddings, a window length of 7 was utilized, while GloVe embeddings were trained with a window length of 10. In the case of GloVe, the training process spanned 30 epochs, employing a learning rate of 0.05 to optimize the model parameters. These values of hyperparameters were arrived after experimentation with several other values.

6 Results

The performance metrics of our models on the test set and development set are presented in Table 2, where "Acc" represents accuracy and "F1" denotes the macro F1 score. Notably, our model demonstrated strong performance on the development set. However, it is worth mentioning that the performance on the test set was comparatively lower. It is important to highlight that our top-performing model utilizes an unsupervised approach leveraging Word2Vec embeddings and cosine similarity.

Despite the varying performance, most of our models consistently outperformed the baseline.

Model Performance on Dev and Test set				
Model	Dev Set		Test set	
	Acc	F1	Acc	F1
Baseline	0.798	0.444	**0.7449**	0.4269
Multi-level approach	0.74	0.65	0.4898	0.4102
Multi-Feature approach	**0.81**	**0.66**	0.6224	0.4966
Word2vec-cosine	0.71	0.62	0.6429	**0.5238**
Word2vec-cosine without replacement	0.62	0.56	0.6020	0.5072
GloVE-cosine	0.64	0.56	0.6020	0.4694
Transformer-cosine	0.60	0.46	0.5612	0.4150
Transformer-euclidean	0.60	0.46	0.5816	0.4421
Transformer-manhattan	0.62	0.49	0.5612	0.4149

Table 2: Performance comparison of all our models

Analysis from Table 2 reveals a notable enhancement in model performance with the replacement of the second-best answer. The subsequent comparison, illustrated in Tables 3 and 4, highlights the impact of this replacement on the Wav2Vec-cosine model's results on both the training and development sets, considering the influence of two distinct similarity scores. Specifically, 'Q' signifies instances where the Question-Answer similarity surpasses the Summary-Answer similarity, while 'S' denotes the reverse scenario. The predictions of models in italics were submitted in Post-evaluation period.

Observing Tables 3 and 4, it becomes evident that the number of accurate predictions substantially increases in the development set, relative to its total size. In the Codalab leader-board we ranked 16 out of 21 teams, and in the overall laeder-board we ranked 15 out 21 teams.

7 Conclusion and Future scope

The dataset presents challenges for models to grasp the intricate legal context, resulting in subpar performance of regular supervised models. Unsupervised models heavily rely on embeddings, but available transformers inadequately capture the dataset's nuances. These models operate under the assumption of at least one correct answer per question; however, instances where all answers were labeled as incorrect hindered unsupervised model performance.

Training Set Counts:		
Higher score	R/W	Count
Q	R	143
Q	W	81
S	R	284
S	W	158
Development Set Counts:		
Higher score	R/W	Count
Q	R	11
Q	W	14
S	R	41
S	W	18

Table 3: Distribution of right (R) and wrong (W) predictions before replacement

Training Set Counts:		
Higher score	R/W	Count
Q	R	144
Q	W	80
S	R	286
S	W	156
Development Set Counts:		
Higher score	R/W	Count
Q	R	14
Q	W	11
S	R	46
S	W	13

Table 4: Distribution of right (R) and wrong (W) predictions after replacement

Future endeavors entail amalgamating these models into a unified super model. This super model would aggregate predictions from various models to yield a singular final prediction, enhancing overall performance and addressing the limitations of individual approaches. An alternative strategy involves leveraging Siamese networks to learn similarity, addressing challenges encountered by unsupervised models when all answers for a particular question are labeled as incorrect (0). By employing Siamese networks, we believe that the model can effectively capture nuanced similarities

between question-answer pairs, and provide better predictions. Exploring other kind of summarizers and using other transformers for summarization such BART (Lewis et al., 2020) may also increase the overall performance of all the systems used in this paper. Data augmentation (Feng et al., 2021) can also be implemented to get better Word2Vec and GloVE embeddings.

Acknowledgements

We would like to thank the organizers, reviewers and SemEval - 2024 Chairs for their valuable insights and helpful suggestions.

References

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Saibo Geng, Rémi Lebret, and Karl Aberer. 2021. Legal transformer models may not always help.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Wei Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *2021 International Conference on Learning Representations*. Under review.

Lena Held and Ivan Habernal. 2024. SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 56–65, Brussels, Belgium. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Interpretable long-form legal question answering with retrieval-augmented large language models.

Jorge Martinez-Gil. 2023. A survey on legal question–answering systems. *Comput. Sci. Rev.*, 48(C).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Neural and Information Processing System (NIPS)*.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA. Omnipress.

Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. 2023. Pre-trained language models for the legal domain: A case study on indian law. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, page 187–196, New York, NY, USA. Association for Computing Machinery.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J. Liu, Sharan Narang, Wei Li, and Yanqi Zhou. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. Technical report, Google.

Mohd Usama, Wenjing Xiao, Belal Ahmad, Jiafu Wan, Mohammad Mehedi Hassan, and Abdulhameed Alelaiwi. 2019. Deep learning based weighted feature fusion approach for sentiment analysis. *IEEE Access*, 7:140252–140260.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Abdelhak at SemEval-2024 Task 9 : Decoding Brainteasers, The Efficacy of Dedicated Models Versus ChatGPT

Abdelhak Kelious
ATILF, University of Lorraine
And CNRS / France
abdelhak.kelious@univ-lorraine.fr

Mounir Okirim
ESIEA, Graduate School
of Engineering / France
okirim@et.esiea.fr

Abstract

This study introduces a dedicated model aimed at solving the BRAINTEASER task 9 (Jiang et al., 2024), (Jiang et al., 2023), a novel challenge designed to assess models' lateral thinking capabilities through sentence and word puzzles. Our model demonstrates remarkable efficacy, securing Rank 1 in sentence puzzle solving during the test phase with an overall score of 0.98. Additionally, we explore the comparative performance of ChatGPT, specifically analyzing how variations in temperature settings affect its ability to engage in lateral thinking and problem-solving. Our findings indicate a notable performance disparity between the dedicated model and ChatGPT, underscoring the potential of specialized approaches in enhancing creative reasoning in AI.

1 Introduction

The BRAINTEASER task (Jiang et al., 2023) aims to challenge the lateral thinking abilities of models, setting it apart from traditional tasks focused on vertical logical reasoning. It introduces lateral thinking puzzles in the form of multiple-choice questions to test the models' ability to think creatively and challenge common sense associations. The goal is to identify the gap between human and model performances in creative thinking, highlighting the need for progress in AI's creative reasoning abilities. NLP (Natural Language Processing) transformer models have revolutionized text understanding and generation with their architecture capable of processing word sequences more efficiently. For multiple-choice questions, these models utilize their ability to understand context and language nuances to select the most appropriate answer from several options. Thanks to deep learning and attention mechanisms, they excel in various NLP tasks, significantly improving the accuracy and relevance of responses generated in complex contexts. The integration of NLP transformer models into the BRAINTEASER task aims to explore their ability to solve lateral thinking puzzles in the form of multiple-choice questions. This approach highlights the challenges posed by deep language understanding and the creativity required to surpass traditional logical reasoning. It emphasizes the importance of advancing in the development of models capable of navigating beyond common sense associations, encouraging innovation in the interpretation and generation of complex and nuanced responses. In our study, we will explore the ability of language models to handle this task, with the following main contributions of this paper :

- Development of a dedicated model for this task with a good result for the sentence puzzle task (Rank 1 in the test phase).
- A comparative analysis with ChatGPT: Specifically, the relationship of temperature with lateral thinking and performance.

2 Shared Task Description

The BRAINTEASER Shared Task 9 is a Question Answering (QA) task based on evaluating the capacity of language models to engage in lateral thinking and to solve puzzles that require unconventional thinking. BRAINTEASER comprises two distinct subtasks: Sentence Puzzle and Word Puzzle, both of which involve defying commonsense "defaults" but through different methodologies.

- Sentence Puzzle: Create sentence-based brain teasers where the challenge lies in interpreting sentence snippets in a way that goes against commonsense expectations.
- Word Puzzle: Design word-based brain teasers that require rethinking the default meanings of words, with a focus on the composition of letters in the target question.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 200–205

Both tasks include an adversarial subset, created by manually modifying the original brain teasers without changing their latent reasoning path. They construct adversarial versions of the original data in two ways:

- (SR) Semantic Reconstruction rephrases the original question without changing the correct answer and the distractors.
- (CR) Context Reconstruction keeps the original reasoning path but changes both the question and the answer to describe a new situational context

Distractors are generated by identifying the implicit and explicit premises of a puzzle and then manually overwriting these premises, ensuring they remain incorrect but challenging.

The BRAINTEASER paper reveals a significant gap between human performances and AI models, and underscores the need to enhance lateral reasoning in language models.

3 Related Work

The task of commonsense reasoning has long been a challenge for deep learning and has been the subject of research for several years, accompanied by various benchmarks such as (Nie et al., 2020), which introduces a new large-scale NLI benchmark dataset created through an adversarial process involving humans and models. This improves NLI models' performance on popular benchmarks and reveals their weaknesses, offering a dynamic framework for continuous improvement in natural language understanding. A study demonstrated a simple and unsupervised method for commonsense reasoning using language models trained on vast text corpora, significantly outperforming state-of-the-art methods on Pronoun Disambiguation Problems and the Winograd Schema Challenge without the need for annotated knowledge bases or manually engineered features (Trinh and Le, 2019).

Transformer models like BERT (Devlin et al., 2019), GPT (Brown et al., 2020), and their variants have revolutionized natural language understanding, including question answering (Qu et al., 2019). Their architecture captures semantic and contextual nuances (Ethayarajh, 2019) (Zhang et al., 2020), proving exceptionally effective in comprehending and responding to complex inquiries. By training on extensive text corpora, they develop a deep understanding, enabling them to identify the most plausible answers among multiple choices (Roy et al., 2023) (Ravi et al., 2023).

Large pretrained language models (PLMs) can achieve near-human performance on commonsense reasoning tasks by generating contrastive explanations that highlight the key attributes needed to justify correct answers. This approach not only improves performance on commonsense reasoning benchmarks but also produces explanations judged by humans as more relevant and understandable (Paranjape et al., 2021)

Recent studies reveal that ChatGPT has notable capabilities to effectively solve a variety of problems in several languages, including the task of answering questions. Moreover, its performance improves with each new version. ChatGPT excels in certain areas but also has its limitations in terms of consistency and complex reasoning tasks.(Tan et al., 2023).

4 Proposed Approach

4.1 Methodology

In our study, we have developed a model based on transformers for multiple-choice questions, where each option is combined with the question to form separate pairs. These pairs are then pre-processed as distinct inputs for the already pre-trained model. The preprocessing includes adding special tokens like [CLS] at the beginning and [SEP] to separate the question from the choice. Each pre-processed question-choice pair is passed through the transformer model, which encodes each pair using its bidirectional attention mechanism, allowing every word in the pair to capture the context of the entire sentence and the related choice. For each question-choice pair, the model generates a feature vector from the output associated with the [CLS] token, which serves as a summary of the information contained in the pair. This means that for a question with four answer choices, the model would be run four times (once for each question-choice pair). This process allows for the consideration of the full context of the question as well as that of each individual answer choice, which is crucial for understanding which choice best answers the question.

The feature vector for each question-choice pair is then passed through a dense (or fully connected) layer, which reduces the vector's dimensionality to a number corresponding to the number of classes or answer categories. After the dense layer, a softmax activation function is applied to convert the scores

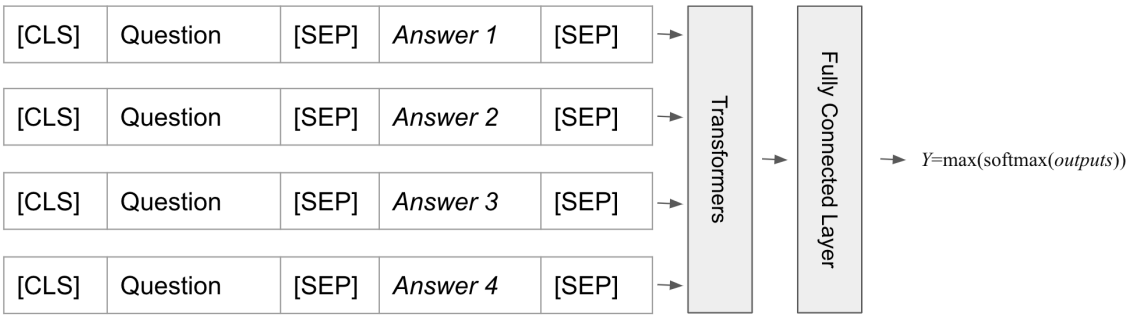


Figure 1: The overall architecture for predicting BRAINTEASER

Results

#	User	Entries	Date of Last Entry	S_ori	S_sem	S_con	S_ori_sem	S_ori_sem_con	S_overall	W_ori	W_sem	W_con	W_ori_sem	W_ori_sem_con	W_overall
2	abdelhak	3	01/25/24	1.000 (1)	1.000 (1)	0.950 (1)	1.000 (1)	0.950 (1)	0.983 (1)	0.625 (10)	0.625 (10)	0.594 (10)	0.562 (12)	0.406 (15)	0.615 (16)

Figure 2: The Ranking Leaderboard Displaying Our Position

into probabilities.

The softmax function is ideal for classification tasks because it transforms the scores into a set of probabilities that sum up to 1, making the scores directly interpretable as the probabilities that each choice is the correct answer. Figure 1 illustrates the prediction process described above.

The prediction formula can be expressed as follows in our model:

Each question-choice pair (Q, C_i) is preprocessed to form an input sequence X_i by concatenating the question Q with each choice C_i and adding special tokens:

$$X_i = [CLS] + Q + [SEP] + C_i + [SEP]$$

The transformer model processes each X_i separately to encode the pair, utilizing its bidirectional attention mechanism. The output for each token in X_i is obtained, but we are specifically interested in the output associated with the [CLS] token, $T_{[CLS],i}$, which captures the contextualized representation of the pair:

$$T_{[CLS],i} = TransformerModel(X_i)$$

The feature vector F_i is extracted from the transformer output associated with the [CLS] token for each question-choice pair:

$$F_i = ExtractFeatureVector(T_{[CLS],i})$$

Each feature vector F_i is passed through a dense layer to reduce its dimensionality to the number of classes N, resulting in a reduced feature vector R_i:

$$R_i = DenseLayer(F_i)$$

The softmax activation function is applied to R_i to convert the scores into probabilities P_i, indicating the likelihood that each choice is the correct answer:

$$P_i = Softmax(R_i) = \frac{e^{R_i}}{\sum_{j=1}^{N} e^{R_j}}$$

Where:

- Q represents the question.
- C_i represents the ith answer choice.
- X_i is the input sequence formed by concatenating Q and C_i with special tokens.
- $T_{[CLS],i}$ is the transformer output for the [CLS] token for the ith question-choice pair.
- F_i is the feature vector extracted from $T_{[CLS],i}$.
- R_i is the reduced feature vector after passing F_i through a dense layer.
- P_i represents the probabilities that each choice C_i is the correct answer, obtained after applying the softmax function to R_i.

4.2 Evaluation Method

The BRAINTEASER task proposes the following evaluation system, each system is evaluated based on the following two accuracy metrics:

Instance-based Accuracy: They consider each question (original/adversarial) as a separate instance. They report accuracy for the original and its adversaries.

Group-based Accuracy: Each question and its associated adversarial instances form a group, and a system will only receive a score of 1 when it correctly solves all questions in the group.

The final score corresponds to the average of all the scores.

4.3 Results

We trained our model using the pre-trained language model DeBERTa-v3-base (He et al., 2023) over 5 learning epochs, with a learning rate of 5e-5 and a batch size of 16. The results obtained are presented in official Leaderboard of the task in the evaluation phase 2.

Our model stands out for its good performance in sentence-type puzzles, ranking first with with an average accuracy score of **0.98** (leaderboard 2) . This means it excels particularly in thinking challenges where the puzzle, often contrary to common sense, is based on sentence excerpts. On the other hand, for word-based puzzles, which require finding a solution that goes against the usual meaning of words by focusing on the letter composition of the posed question, our model shows lower performance. It ranks 16th with a total score of **0.61** . This performance difference suggests that, although our model is very skilled at solving puzzles involving the understanding and manipulation of sentences, it could benefit from improvement in the area of word-based puzzles. This indicates an opportunity to deepen our research and development efforts on word-type puzzles to enhance the versatility and overall effectiveness of our model.

5 ChatGPT Analysis

5.1 Zero-shot Predictions

Given that we are currently in the era of ChatGPT, it's challenging to approach our study without including a comparison to evaluate the role of this task in relation to ChatGPT. We crafted a simple and explicit prompt with ChatGPT turbo 3.5 on February 5, 2024, assessing ChatGPT's logical reasoning ability using various prompts in a qualitative manner. However, we faced challenges in determining the optimal prompt, as the same input does not always lead to the desired output. Hallucinations related to conversation history were resolved by initiating a new session for each iteration. In the end, we settled on the following prompt:

> “””
> Question ?
> list of choices :
> 1- Answer 1.
> 2- Answer 2.
> 3- Answer 3.
> 4- Answer 4.
> Response should be in json format :
> { “answer”: Number of the choice }
> “””

We achieved a total score of **0.59** for the sentence-puzzle task and **0.27** for the word-puzzle task, scores that do not necessarily match the expected performance for a model like ChatGPT. This suggests that, although ChatGPT was not specifically trained for this task, it might not be able to compete with models that were specially designed for it. ChatGPT was trained on a vast dataset, but it is assumed that most of this data is well-structured and more aligned with linear thinking rather than lateral thinking, which explains its moderate performance in this area.

5.2 The Effect of Temperature

The temperature parameter in language models for natural language processing is a hyperparameter used to control the diversity of predictions made by the model during text generation. Temperature adjusts the likelihood of predictions based on their calculated probability, thereby influencing the level of risk or surprise in the choice of generated words. Adjusting the temperature allows for control over the trade-off between creativity and safety in text generation. Finding the right temperature depends on the specific application, the domain of use, and preferences for the balance between innovation and reliability in the generated responses. A low temperature close to 0 produces more conservative and repetitive responses, while a high temperature close to 1 yields more varied and creative responses.

Is there a relationship between temperature and lateral thinking ? Although the temperature

setting in language models and lateral thinking operate in different domains, they share a common goal of fostering creativity and innovation by breaking conventions and exploring possibilities beyond those that are immediately obvious. Lateral thinking encourages questioning assumptions and considering a variety of different perspectives. Similarly, by adjusting the temperature to favor less likely word selections, a language model can "think" more laterally, exploring linguistic options that would not be considered at a lower temperature. Therefore, we will measure the performance of ChatGPT based on temperature, relationship between temperature and lateral thinking.We will launch several runs by increasing the temperature from 0 to 1.2

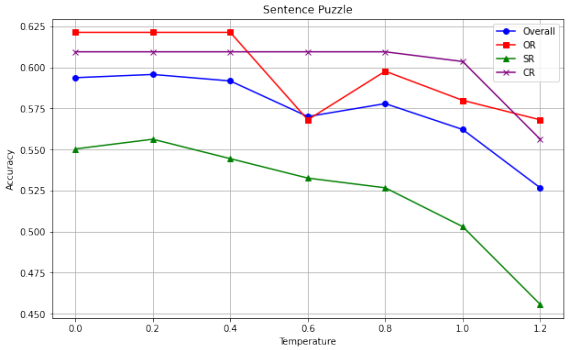


Figure 3: ChatGPT Performance Across Different Temperatures (Sentence puzzle)

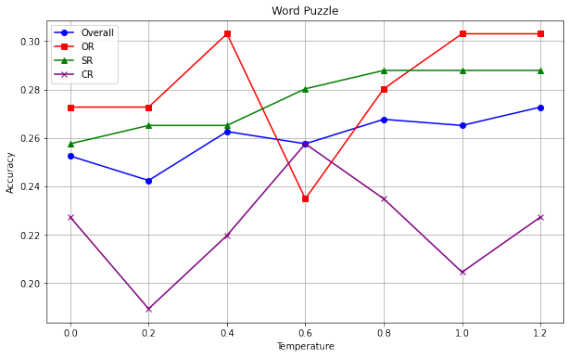


Figure 4: ChatGPT Performance Across Different Temperatures (Word puzzle)

Sentence Puzzle : The graphic 3 represents four data series corresponding to different test scenarios for the sentence puzzle task: Overall, OR (Original), SR (Semantic Reconstruction), and CR (Context Reconstruction). "Overall" indicates a benchmark or an overall average of performance, while OR shows stable results, suggesting a consistent baseline. CR follows a trend similar to OR, indicating that contextual reconstruction performs comparably to the original. In contrast, SR shows a notable degradation in performance towards the end, which could suggest that the semantic reconstruction method is less stable or effective under certain conditions. The data set suggests that while OR and CR methods maintain a degree of consistency, SR might involve a riskier or more innovative approach, which could be likened to a "higher temperature" in the context of lateral thinking, leading to more varied and potentially less predictable outcomes. However, increasing the temperature does not allow the model to perform better on a task, on the contrary, performance decreases.

Word puzzle : In the case of word puzzles 4, it is difficult to conclude as there are no clear trends observed. However, for the overall general case, it is noted that performance increases very slightly with temperature, which stands out in comparison to the sentence puzzle task, potentially because word puzzles better illustrate lateral thinking. In this case, the focus is not on the sentence, which contains more semantic aspects. The answer in this task violates the default meaning of the word and focuses on the letter composition of the target question.

6 Conclusion

Our research underscores the significance of dedicated models in advancing AI's capability to solve complex lateral thinking tasks, as exemplified by our model's top-ranking performance in the BRAINTEASER sentence puzzles. The comparative analysis with ChatGPT highlights the limitations of general-purpose models in specific creative reasoning challenges, despite their overall versatility. The study also reveals the nuanced role of temperature settings in modulating ChatGPT's performance, offering insights into optimizing AI models for enhanced creativity and lateral thinking. Future work should focus on bridging the gap in word puzzle performance and further refining the balance between creativity and logical reasoning in AI systems.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot

learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2023. Brainteaser: Lateral thinking puzzles for large language model. *arXiv preprint arXiv:2310.05057*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Prompting contrastive explanations for commonsense reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4179–4192, Online. Association for Computational Linguistics.

Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. Bert with history answer embedding for conversational question answering. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 1133–1136.

Sahithya Ravi, Aditya Chinchure, Leonid Sigal, Renjie Liao, and Vered Shwartz. 2023. Vlc-bert: visual question answering with contextualized commonsense knowledge. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1155–1165.

Pradeep Kumar Roy, Sunil Saumya, Jyoti Prakash Singh, Snehasish Banerjee, and Adnan Gutub. 2023. Analysis of community question-answering issues via machine learning and deep learning: State-of-the-art review. *CAAI Transactions on Intelligence Technology*, 8(1):95–117.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of chatgpt as a question answering system for answering complex questions. *arXiv preprint arXiv:2303.07992*.

Trieu H. Trinh and Quoc V. Le. 2019. A simple method for commonsense reasoning.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.

OUNLP at SemEval-2024 Task 9: Retrieval-Augmented Generation for Solving Brain Teasers with LLMs

Vineet Saravanan
Cranbrook Schools
Bloomfield Hills, MI
vineetsaravanan@gmail.com

Steven Wilson
Oakland University
Rochester, MI
stevenwilson@oakland.edu

Abstract

The advancement of natural language processing has given rise to a variety of large language models (LLMs) with capabilities extending into the realm of complex problem-solving, including brainteasers that challenge not only linguistic fluency but also logical reasoning. This paper documents our submission to the SemEval 2024 Brainteaser task, in which we investigate the performance of state-of-the-art LLMs, such as GPT-3.5, GPT-4, and the Gemini model, on a diverse set of brainteasers using prompt engineering as a tool to enhance the models' problem-solving abilities. We experimented with a series of structured prompts ranging from basic to those integrating task descriptions and explanations. Through a comparative analysis, we sought to determine which combinations of model and prompt yielded the highest accuracy in solving these puzzles. Our findings provide a snapshot of the current landscape of AI problem-solving and highlight the nuanced nature of LLM performance, influenced by both the complexity of the tasks and the sophistication of the prompts employed. All the code, along with the data used, is available on our GitHub[1]

1 Introduction

The pursuit of creating artificial intelligence models with advanced reasoning and problem-solving capabilities has led researchers down the path of deploying brainteasers as a benchmark for AI systems' linguistic and reasoning prowess. These brainteasers are more than trivial or recreational challenges; they are testaments to the complexity of human cognition, embedding layers of semantics, pragmatics, and world knowledge that remain elusive to AI systems. The gulf between the operational logic of current AI models and the intricate understanding displayed by the human mind is significant, particularly in domains necessitating advanced reasoning and a robust common sense foundation. This disparity is not only observed but keenly felt in the context of AI systems' interaction with human language and thought (Mahowald et al., 2023).

The limitations of pattern recognition as the mainstay of AI systems' learning mechanisms have been critically examined, sparking a discourse that emphasizes the imperative for AI systems to transcend these confines. Rigorous benchmarks that challenge AI systems to demonstrate inferential reasoning are essential to catalyze this evolution (Sawada et al., 2023). Brainteasers emerge as one medium through which AI systems' competencies can be evaluated. They are not simply puzzles to be solved but are reflective of the complex, often ambiguous nature of human communication and problem-solving.

The BRAINTEASER task introduced at SemEval 2024 (Jiang et al., 2024) is part of this evolution of AI system assessment, standing at the center of linguistic analysis and computational intelligence. It is designed to evaluate what machines can understand and how they can apply this understanding in a manner similar to human thought processes. Language models, such as GPT-3.5[2] and GPT-4 (OpenAI, 2023), are increasingly being subjected to these tests to gauge their mastery over language and logic, as demonstrated in recent comparative analyses (Espejel et al., 2023). The BRAINTEASER task's format, which intertwines linguistic cues with logical conundrums, requires systems to not only comprehend the text at a superficial level but to delve into the implied, the inferred, and the intuitive aspects that are second nature to human beings.

By benchmarking language models against brainteasers within the framework of the BRAIN-

[1] https://github.com/VSPuzzler/OUNLP-at-SemEval-2024-Task-9

[2] https://openai.com/blog/chatgpt

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 206–212

TEASER task, we are able to learn more about the current capabilities of popular LLMs. This work can help to provide a direction for future research by pinpointing where current models fall short and where the next wave of innovation is urgently needed.

Our approach involved testing different LLM models. We web-scraped example riddles and used them as an example for the model. Additionally, we tested with the closest riddle and the most different riddle and found that GPT-4.0 oneshot with a similar riddle worked best for the Sentence Puzzle and GPT-4.0 oneshot with a different riddle worked best for the Word Puzzle. The Word Puzzle turned out to be a significantly harder task than the Sentence Puzzle.

2 Related Work

The exploration of reasoning abilities in large language models (LLMs) has been the focus of several studies in recent years. Notably, work by OpenAI provides foundational insights into the capabilities of GPT-3, especially highlighting its potential in solving reasoning tasks through few-shot learning (Brown et al., 2020). This work is particularly relevant as it demonstrates how providing a few examples can significantly improve an LLM's ability to solve reasoning problems, akin to the one-shot and few-shot techniques examined in our study.

Furthering the discussion on reasoning, work has been done that discusses the 'chain-of-thought' (CoT) prompting method, where models are guided to articulate intermediate steps when solving complex tasks (Wei et al., 2022). This process is similar to the explanation method in solving the brainteasers, which encourages models to elaborate on their reasoning, leading to improved performance.

The brainteasers in the training data provided often require making analogies and similarities in reasoning. Work has been done that offers an analysis of how word embeddings capture semantic relationships, which can be fundamental in retrieving similar examples to aid reasoning (Allen and Hospedales, 2019). This is directly linked to our one-shot similar and few-shot in-context learning approaches, where the ability of an LLM to use analogous examples influences its problem-solving effectiveness.

Moreover, the strategies for solving brainteasers with AI systems have been enriched by incorporating external knowledge bases. An investigation has been conducted into the inherent knowledge within language models and their ability to function as knowledge bases (Petroni et al., 2019). The integration of external knowledge is particularly pertinent to tasks requiring common sense and real-world information, underscoring the importance of knowledge retrieval in the context of a brainteaser. Lastly, a pivotal study has been done that introduces a dataset designed to probe AI systems' common sense reasoning capabilities (Talmor et al., 2018). This study aligns with our aim in solving brainteasers to evaluate the capacity of LLMs to handle questions that necessitate an understanding of the world as humans perceive it.

3 Methodology

Our experimental design relies on prompt engineering to explore the effectiveness of language models in solving brainteasers. In this study, we experimented with different prompt structures to determine their impact on the model's performance. The primary prompt format tested was structured as follows: "Please pick the best choice for the brain teaser. Each brain teaser has only one possible solution, including the choice 'none of the above.' The answer should only provide the choice text." This directive was chosen to explicitly instruct the model to select a single, most appropriate answer from a set of given options. To ensure a controlled variable, we explicitly presented the model with the choices, observing how it navigates the selection process when options are directly provided.

An interesting observation was made regarding the specification of the type of brainteaser. Initially, it was hypothesized that indicating whether the puzzle was a 'word puzzle' or a 'sentence puzzle' would aid the models in narrowing down their reasoning scope, thereby improving accuracy. However, the results indicated that such specifications did not significantly affect the models' performance. This finding suggests that the models possess a level of task generalization, wherein they apply similar reasoning processes to both types of puzzles without the need for explicit differentiation.

Furthermore, we explored the effect of including choices within the prompt. By contrasting scenarios with and without provided options, we aimed to assess whether the presence of choices would guide the model to a correct answer more efficiently. Prompts structured to request the model

to pick from provided choices explicitly did not significantly alter the success rate compared to when no choices were given. This aspect of the study aimed to discern the degree to which the models rely on contextual clues versus intrinsic problem-solving capabilities. For example, in a test with a bad prompt on GPT-3.5 without choice, a 27.60% accuracy was achieved, while with the same prompt with choices, a 28.80% accuracy was achieved. Due to the slight increase in accuracy, we decided to include choices for the rest of the prompts used.

We also implemented explanation and chain of thought (CoT) reasoning to guide the language model toward a more structured and reasoned approach when tackling brain teasers. Explanation prompts encouraged the model to articulate the rationale behind its chosen answers. Similarly, CoT prompts aimed to simulate a step-by-step reasoning process, mirroring how humans might approach problem-solving.

To further enhance the accuracy of our language model in solving brain teasers, we adopted a one-shot in-context learning approach, leveraging a large dataset of riddles as context for the model. We extracted a comprehensive collection of 3,899 riddles by downloading texts from the `riddles.com` website, including both the questions and their corresponding answers. This dataset was a reference for the model to draw upon when presented with new puzzles.

We employed two distinct strategies for selecting a relevant riddle from this dataset to use as an example in our one-shot method. The first strategy aimed to identify the riddle most *similar* to the brain teaser in question, believing that a similar context might prime the model more effectively for the task at hand. Conversely, the second strategy sought out the riddle most *dissimilar* to the brain teaser, hypothesizing that a contrasting example could stimulate a broader range of the model's reasoning capabilities.

To facilitate the rapid identification of the most similar or dissimilar riddle, we encoded each riddle question into a vector representation using the `all-MiniLM-L6-v2 model` (Reimers and Gurevych, 2019) in the `SentenceTransformers`[3] Python package. This allowed us to compute the cosine similarity between the vector representation of the new brain teaser and those of the riddles in our dataset. Using this similarity score, we could efficiently identify the riddle that was either closest or farthest in semantic space from the brain teaser presented to the model. The highest cosine similarity is used as the similar riddle, while the lowest cosine similarity is used as the dissimilar riddle.

This approach significantly improved the efficiency of selecting an appropriate example riddle, enabling a more streamlined integration of the one-shot learning method into our experimental setup. The rationale behind using cosine similarity was to leverage the high-dimensional space in which language representations reside, making it possible to quantify the semantic proximity between different textual inputs effectively. A depiction of the different elements of different prompts is depicted in Figure 1.

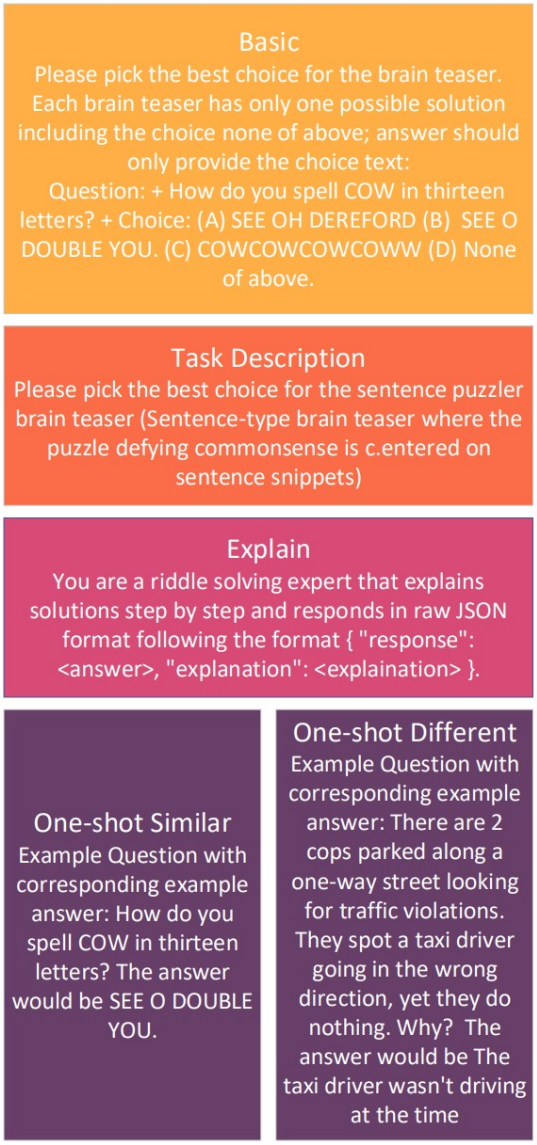


Figure 1: Example prompts tested

[3] https://sbert.net

In an effort to enhance the accuracy of solutions provided by the language model for word puzzles, we also experimented with an innovative approach modeled after collective human problem-solving dynamics. This method involved simulating a "council" of three hypothetical individuals engaged in a discussion about a puzzle, with the aim of reaching a consensus on the answer.[4] The intent was to emulate the collaborative approach often used in human group problem-solving, where different perspectives and thought processes can lead to more accurate solutions. The implementation of this method required the model to generate three distinct responses, each purportedly from a different "council member," who would then "discuss" their reasoning and perspectives on the puzzle. Following this simulated deliberation, the model was prompted to synthesize the viewpoints into a single, collective answer (see example in Appendix A). Despite the creative nature of this technique, the results were not as promising as anticipated. The accuracy of the word puzzles did not show significant improvement using the council-based discussion method. This outcome suggests that while the approach mirrors human group interactions, it may not translate effectively within the constraints of a single AI model's processing capabilities.

4 Results

We ran and tested the LLMs on the training set (Jiang et al., 2023) with each combination of model and prompt. This allows us to get a comprehensive view of the performance of the LLMs across different prompts. It is important to increase the general accuracy across all LLMs by adding more information about the question to the prompt, along with examples. The models we evaluated included GPT-3.5, GPT-4, Gemini Pro (Team et al., 2023), and a suite of memory-efficient language models from the `languagemodels` repository.[5] The models used from this package included neural-chat-7b-v3-1, flan-alpaca-xl, flan-alpaca-gpt4-xl, flan-t5-xl, fastchat-t5-3b-v1.0, LaMini-Flan-T5-783M, flan-t5-large, LaMini-Flan-T5-248M, flan-alpaca-base, flan-t5-base, dialogstudio-t5-base-v1.0, LaMini-Flan-T5-77M, flan-t5-small, phi-1_5, LaMini-GPT-774M, and LaMini-GPT-124M. This set explores sixteen models, yet they never outperformed the other LLMs (GPT and Gemini series). Therefore, we only report the results from the best of these models for each prompt and task in the final result tables, Table 1 and Table 2.

Prompt	GPT-3.5	GPT-4.0	Gemini	`languagemodels`
basic	0.288	0.649	**0.803**	0.359
task desc.	0.477	0.645	**0.753**	0.383
+ CoT	**0.722**	0.692	0.671	0.314
+one-shot sim.	0.650	**0.809**	0.753	0.633
+one-shot diff.	0.680	**0.825**	0.759	0.345
+one-shot sim. + CoT	**0.710**	0.686	0.637	0.686
+one-shot diff. + CoT	0.670	**0.704**	0.655	0.347

Table 1: Accuracy of LMs using different prompts on the **Sentence Puzzle** task. **Bold** indicates the best model for a given prompting strategy, and **underlined** indicates the best overall approach for the task. The `languagemodels` column shows the best score achieved by any model from the `languagemodels` library.

Prompt	GPT-3.5	GPT-4.0	Gemini	`languagemodels`
basic	0.346	0.508	**0.531**	0.341
task desc.	0.341	0.487	**0.494**	0.354
+ CoT	0.520	**0.641**	0.351	0.323
+one-shot sim.	0.485	**0.649**	0.530	0.553
+one-shot diff.	0.470	**0.621**	0.505	0.356
+one-shot sim. + CoT	**0.553**	0.540	0.384	0.242
+one-shot diff. + CoT	0.513	**0.586**	0.354	0.333

Table 2: Accuracy of LMs using different prompts on the **Word Puzzle** task. **Bold** indicates the best model for a given prompting strategy, and **underlined** indicates the best overall approach for the task. The `languagemodels` column shows the best score achieved by any model from the `languagemodels` library.

It was found that the one-shot method consistently had the top 2 accuracy in the prompts studied, proving the efficiency of one-shot methods for LLMs in general. Gemini's accuracy when only using the basic prompt was very high for the Sentence Puzzle task, which shows Gemini's versatility and adaptability to different questions with high accuracy without needing examples to perform well. The GPT-4 system with the basic prompt with a chain-of-thought method also proved to be highly accurate.

The Chain of Thought approach has been shown

[4] https://github.com/dave1010/tree-of-thought-prompting

[5] https://github.com/jncraton/languagemodels

to improve accuracies for LLMs. Despite these efforts, we observed that this strategy did not lead to a measurable increase in accuracy. This outcome suggests that while such prompts can often lead to more interpretable answers, they do not necessarily enhance the model's ability to deduce the correct solution in the context of brain teasers. Further research may explore whether the complexity of the puzzles or the inherent limitations of the models' understanding contributed to this result.

Despite the overall increase in accuracy observed with more informative prompts, GPT-4 did not always outperform GPT-3.5 with prompts like the task description + explain. This highlights that while advancements in model architecture contribute to enhanced performance, they do not guarantee superior outcomes in every scenario, particularly in specialized tasks like puzzle-solving. Since GPT-4 is trained on a larger range of data sources to improve general performance across a broad range of tasks, the generalized training approach may lead GPT-4 not performing as well in this specific task.

Furthermore, we found that employing chain-of-thought and explanation methodologies did not significantly improve performance in this context. This deviation from expected outcomes may indicate that for certain types of puzzles, these approaches do not align with the models' strengths or the nature of the problem-solving process required.

The performance of open-source packages like `languagemodels` was notably lower compared to their commercial counterparts. This gap underscores the developmental distance that open-source models need to traverse to reach the sophistication level of models like GPT-4 or Gemini, suggesting that access to extensive datasets, computing resources, and proprietary algorithms plays a significant role in model performance. Typically the models from this set that worked best were the LaMini-Flan-T5 class, which was always the case for the one-shot setting. The main exceptions to this were in the zero-shot scenario, specifically with the basic prompt (neuralchat worked best for Sentence Puzzle and dialogstudio worked best for Word Puzzle) and the task description prompt (phi-1.5 for SP and neuralchat for WP). These cases provide positive examples of situations in which lightweight, open models are more competitive with proprietary, closed models.

Additionally, the increased difficulty of word puzzles presents a notable challenge, potentially due to their reliance on nuanced understanding, cultural context, and semantic associations that can be challenging even for human solvers. This complexity is reflected in the lower accuracy rates across all models for word puzzles when compared to sentence puzzles, implying that word puzzles may represent a closer analog to human-level problem-solving and, as such, provide a more stringent test of AI reasoning and language capabilities.

The three submission prompts submitted are GPT-4 one-shot different for Sentence Puzzle and GPT-4 one-shot similar for word puzzles, GPT-4 one-shot similar for Sentence Puzzle and GPT-4 one-shot different for word puzzles, and Gemini basic for Sentence Puzzle and GPT-4 basic + CoT for word puzzle. The first submission received an accuracy score of 0.925 for the sentence puzzle and 0.9375 for the word puzzle, the second a score of 0.95 for the sentence puzzle and 0.78125 for the word puzzle, and the third a score of 0.625 for the sentence puzzle and 0.46875 for the word puzzle. Not that these accuracies are from the "Original" riddles. The second submission was the highest and ranked us 14th in terms of average score, 11th on the Word Puzzle task, and 10th on the Sentence Puzzle task under the name `vspuzzler`. Our submitted system performed exceptionally well on the "Original" version of the brainteasers (ranking 3rd overall for Sentence Puzzle and 7th for Word Puzzle within this subcategory) but underperformed on the "Context Reconstruction" variations of the brainteasers in which the original reasoning path was used within a new situational context.

5 Conclusion

This study's evaluation of LLMs across a range of prompt types provides insights into the strengths and limitations of current AI systems in solving brain teasers. Our findings revealed that in-context learning methods are highly effective, particularly for the Gemini model and GPT-4, when solving sentence puzzles. However, GPT-4 did not consistently outperform GPT-3.5 across all prompt types, which suggests that the latest models do not always guarantee an improvement in task-specific performance. The chain-of-thought and explanation strategies, while enhancing interpretability, did not necessarily translate into higher accuracy, indicating the need for further research into how these models process complex language tasks. The performance gap between proprietary and open-source

models highlights the significant role of resources and proprietary technology in developing LLMs. The increased difficulty of word puzzles suggests that tasks requiring a nuanced understanding and cultural context remain challenging for AI systems, closely mirroring the complexity of human cognition. This study underscores the importance of tailored prompting strategies to leverage the capabilities of LLMs and the potential for future advancements in AI-based problem-solving.

References

Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings. In *International Conference on Machine Learning*, pages 223–231. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jessica López Espejel, El Hassane Ettifouri, Mahaman Sanoussi Yahaya Alassan, El Mehdi Chouham, and Walid Dahhane. 2023. Gpt-3.5, gpt-4, or bard? evaluating llms reasoning ability in zero-shot setting and performance boosting through prompts. *Natural Language Processing Journal*, 5:100032.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. 2023. Dissociating language and thought in large language models: a cognitive perspective. *arXiv preprint arXiv:2301.06627*.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tomohiro Sawada, Daniel Paleka, Alexander Havrilla, Pranav Tadepalli, Paula Vidas, Alexander Kranias, John J Nay, Kshitij Gupta, and Aran Komatsuzaki. 2023. Arb: Advanced reasoning benchmark for large language models. *arXiv preprint arXiv:2307.13692*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

A Appendix

Chain of thought was used as one of the prompts. Below is an example of it in use:

> **Prompt:**
>
> Please pick the best choice for the sentence puzzle brain teaser (Sentence-type brain teaser where the puzzle defying commonsense is centered on sentence snippets). Each brain teaser has only one possible solution including the choice none of above, answer should only provide the choice:
>
> Question: Mr. and Mrs. Mustard have six daughters and each daughter has one brother. But there are only 9 people in the family, how is that possible? Choice:
>
> (A) Some daughters get married and have their own family.
>
> (B) Each daughter shares the same brother.
>
> (C) Some brothers were not loved by family and moved away.
>
> (D) None of above.
>
> **Answer:**

{"response": "B",

"explanation": "Each daughter has one brother, which means there is only one brother shared among all six daughters. If Mr. and Mrs. Mustard have six daughters and one son, together with the two of them, that makes a total of nine family members. Therefore, the correct answer is choice B." }

A council method was also used as one of the prompts. Below is an example of it in use:

Prompt:

Please pick the best choice for the sentence puzzle brain teaser (Sentence-type brain teaser where the puzzle-defying commonsense is centered on sentence snippets). Each brain teaser has only one possible solution including the choice none of above, answer should only provide the choice. Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realises they're wrong at any point then they leave.

Question:Mr. and Mrs. Mustard have six daughters and each daughter has one brother. But there are only 9 people in the family, how is that possible? Choice:

(A) Some daughters get married and have their own family.

(B) Each daughter shares the same brother.

(C) Some brothers were not loved by family and moved away.

(D) None of above.

Answer:

{"response": "B",

"Expert1Thinking": "If there are six daughters and each has one brother, it would initially seem like there should be six brothers. However, the question is phrased in a way that suggests that each daughter has the same single brother.",

"Expert2Thinking": "Taking into account that there are Mr. and Mrs. Mustard, the six daughters can collectively have only one brother to make the total count nine. This brother would be shared among all sisters.",

"Expert3Thinking": "It's a puzzle meant to trick the reader into thinking there are more people than there actually are. The family must consist of Mr. and Mrs. Mustard, their six daughters, and one son who is the brother of all daughters. So, there's only one brother, not six." }

NLP-LISAC at SemEval-2024 Task 1: Transformer-based approaches for Determining Semantic Textual Relatedness

Abdessamad Benlahbib[1] , **Anass Fahfouh**[1], **Hamza Alami**[1] , **Achraf Boumhidi**[1]
[1] LISAC Laboratory, Faculty of Sciences Dhar EL Mehraz, USMBA, Fez, Morocco
abdessamad.benlahbib@usmba.ac.ma , anassfahfouh@gmail.com,
hamza.alami5@usmba.ac.ma, achraf.boumhidi@usmba.ac.ma

Abstract

This paper presents our system and findings for SemEval 2024 Task 1 Track A Supervised Semantic Textual Relatedness. The main objective of this task was to detect the degree of semantic relatedness between pairs of sentences. Our submitted models (ranked 6/24 in Algerian Arabic, 7/25 in Spanish, 12/23 in Moroccan Arabic, and 13/36 in English) consist of various transformer-based models including MARBERT-V2, mDeBERTa-V3-Base, DarijaBERT, and DeBERTa-V3-Large, fine-tuned using different loss functions including Huber Loss, Mean Absolute Error, and Mean Squared Error.

1 Introduction

Semantic Textual Relatedness (STR) is a natural language processing (NLP) task that focuses on measuring the degree of semantic relatedness between two pieces of text. Unlike tasks such as Semantic Textual Similarity (STS), which specifically assess the degree of similarity between texts, STR considers a broader notion of relatedness, encompassing various types of semantic relationships between words, phrases, or sentences.

The goal of STR is to quantify how closely related two pieces of text are in terms of their underlying meaning or semantic content. This relatedness can encompass a wide range of semantic relationships, including:

- **Synonymy**: Words or phrases that have similar meanings.
- **Hyponymy/Hypernymy**: Hierarchical relationships where one word is a more specific instance (hyponym) or a more general category (hypernym) of another word.
- **Meronymy/Holonymy**: Meronymy is a semantic relation between a meronym denoting a part and a holonym denoting a whole.
- **Antonymy**: Words with opposite meanings.
- **Entailment**: One statement logically implies another statement.
- **Association**: Words or concepts that are commonly associated with each other.

In the context of STR, annotators or models are typically presented with pairs of text and asked to judge the degree of relatedness based on the presence of shared concepts or semantic associations. Annotators might provide relatedness scores or labels indicating the strength of the relationship between text pairs.

In this paper, we present our findings on SemEval 2024 Task 1 Track A: Supervised Semantic Textual Relatedness (Ousidhoum et al., 2024b). Our method consists of various transformer-based approaches (Vaswani et al., 2017) fine-tuned using different loss functions including Huber Loss, Mean Absolute Error, and Mean Squared Error.

The rest of the paper is structured in the following manner: Section 2 provides the main objective of the Task. Section 3 describes our system. Section 4 details the experiments. And finally, Section 5 concludes this paper.

2 Task Description

This task aims to predict the semantic textual relatedness (STR) of pairs of sentences across 14 different languages. Participants will rank sentence pairs based on their semantic closeness, ranging from 0 (completely unrelated) to 1 (maximally related), as determined manually. Teams can submit entries for one, two, or all of the following tracks:

- **Track A: Supervised**: Participants are required to submit systems trained using provided labeled training datasets. They may utilize publicly available datasets, but must disclose additional data used and assess its impact on results.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 213–217

- **Track B: Unsupervised**: Participants must submit systems developed without using labeled datasets on semantic relatedness or similarity between text units longer than two words in any language. However, the use of unigram or bigram relatedness datasets from any language is allowed.
- **Track C: Cross-lingual**: Participants must submit systems developed without labeled semantic similarity or relatedness datasets in the target language, but may use labeled dataset(s) from at least one other language. Note: Utilizing labeled data from another track is mandatory for submissions to this track.

3 System Description

To tackle the SemEval 2024 Task 1 Track A: Supervised Semantic Textual Relatedness, we fine-tuned several transformer-based models on an augmented training dataset and with different loss functions including Huber Loss, Mean Absolute Error, and Mean Squared Error. The different steps of our system are described as follows:

- We combined the training and development sets separately for each language in which we participated in. Besides, we duplicated the obtained datasets input, but we shifted the pairs order and we kept the same semantic relatedness score. Table 1 and 2 depict an example of augmenting the English training set.
- We replaced the newline character **\n** with **[SEP]** token in order to separate the input pairs. For example, this input: "Then, in twenty minutes, gather at the runway. \n gathering on the runway, in 20 minutes." will be converted to "Then, in twenty minutes, gather at the runway. [SEP] gathering on the runway, in 20 minutes."
- We tokenized the data using tokenizers associated with the fine-tuned transformer based models.
- We fine-tuned MARBERTv2 (Abdul-Mageed et al., 2021) on the Algerian Arabic data, DarijaBERT (Gaanoun et al., 2024) on the Moroccan Arabic data, DeBERTa-V3-Large (He et al., 2021a,b) on the English data, and mDeBERTa-V3-Base (He et al., 2021a) on the Spanish data.

In the context of semantic textual relatedness tasks, the choice of loss function plays a critical role in guiding the training process and optimizing model performance. Given the diverse nature of textual data and the wide range of semantic relationships to be captured, employing a variety of loss functions can offer several advantages.

Firstly, the Huber Loss function provides robustness to outliers by combining the advantages of Mean Absolute Error (MAE) and Mean Squared Error (MSE). MAE, which calculates the average absolute difference between predicted and target values, is less sensitive to outliers compared to MSE, which squares the differences. By behaving like MSE for large errors and like MAE for small errors, Huber Loss ensures that the training process is less influenced by outliers, thereby enhancing the model's ability to generalize to unseen data.

Secondly, Mean Absolute Error (MAE) serves as a straightforward and intuitive loss function that penalizes deviations from the target scores equally, irrespective of their direction. In tasks such as semantic textual relatedness, where the goal is to predict similarity scores between sentence pairs, MAE provides a direct measure of the magnitude of errors, facilitating easy interpretation and evaluation of model performance.

Lastly, Mean Squared Error (MSE) emphasizes the importance of accurately predicting similarity scores by penalizing larger errors more severely than smaller errors. In scenarios where precise estimation of the degree of relatedness between sentence pairs is crucial, MSE can effectively guide the training process towards minimizing the squared differences between predicted and target values, thereby optimizing model performance.

By leveraging a combination of these loss functions during the fine-tuning process, we aim to capitalize on their respective strengths and enhance the robustness and effectiveness of our models in capturing semantic relationships within textual data. This approach enables us to optimize model performance across various linguistic contexts and achieve competitive results in tasks requiring accurate assessment of semantic textual relatedness.

The decision to augment the data was validated through fine-tuning the models on concatenated train and dev sets, as well as on concatenated train and dev sets with pair shifting. Interestingly, our analysis revealed that pair shifting significantly enhanced the results on the development sets.

PairID	Text	Score
ENG-train-0047	Then, in twenty minutes, gather at the runway. \n gathering on the runway, in 20 minutes.	0.97
ENG-dev-0010	Meat is dropped into a pan. \n A woman is putting meat in a pan.	0.73

Table 1: Sample of the English training set after combining both training and development sets

PairID	Text	Score
ENG-train-0047	Then, in twenty minutes, gather at the runway. \n gathering on the runway, in 20 minutes.	0.97
ENG-dev-0010	Meat is dropped into a pan. \n A woman is putting meat in a pan.	0.73
ENG-train-0047-shifted	gathering on the runway, in 20 minutes. \n Then, in twenty minutes, gather at the runway.	0.97
ENG-dev-0010-shifted	A woman is putting meat in a pan. \n Meat is dropped into a pan.	0.73

Table 2: Sample of the English training set after combining both training and development sets and after shifting the pairs

4 Experimental Results

We experimented our model on the SemEval 2024 Task 1: Semantic Textual Relatedness (STR) test set (Ousidhoum et al., 2024a). The experiment has been conducted in Kaggle environment[1], The following libraries: Transformers - Hugging Face[2] (Wolf et al., 2020), and Keras[3] were used to train and to assess the performance of our models.

4.1 Datasets

Each instance in the training, development, and test sets (Ousidhoum et al., 2024a) is a sentence pair. The instance is labeled with a score representing the degree of semantic textual relatedness between the two sentences. The scores can range from 0 (maximally unrelated) to 1 (maximally related). Figure 1 depicts the training, dev and test sets distributions for Algerian Arabic, Moroccan Arabic, English and Spanish.

The datasets are available via GitHub[4]

4.2 Experimental Settings

We conducted numerous experiments on the development set to obtain the ideal number of epochs and identify the most effective loss function for fine-tuning each model. This paper presents the hyperparameters that yielded the best results on the development set across the target languages:

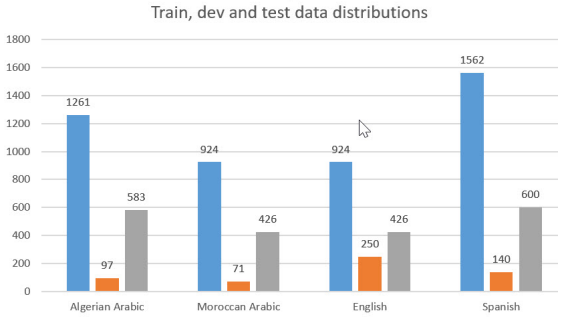


Figure 1: Train, development and test sets distributions for Algerian Arabic, Moroccan Arabic, English and Spanish

- **Algerian Arabic**: We fine-tuned MARBERTv2 using 12 epochs, a maximum sequence length of 200, and Mean Absolute Error as the loss function.

- **Moroccan Arabic**: We fine-tuned DarijaBERT using 5 epochs, a maximum sequence length of 200, and Huber loss as the loss function.

- **English**: We fine-tuned DeBERTa-V3-Large using 5 epochs, a maximum sequence length of 150, and Huber loss as the loss function.

[1] https://www.kaggle.com/
[2] https://huggingface.co/docs/transformers/index
[3] https://keras.io/
[4] https://github.com/semantic-textual-relatedness/Semantic_Relatedness_SemEval2024

- **Spanish**: We fine-tuned mDeBERTa-V3-Base using 12 epochs, a maximum sequence length of 200, and Mean Squared Error as the loss function.

The same parameters were utilized during the final submission phase. Additionally, Table 3 displays the additional hyperparameter settings employed during the fine-tuning process for all models.

Hyperparameters	Settings
Learning rate	10^{-5}
Batch size	4
Optimizer	Adam (Kingma and Ba, 2015)

Table 3: Hyperparameters settings for the model in the experiments

4.3 System Performance

Table 4 depicts the results of our proposed approaches on SemEval 2024 Task 1 Track A Supervised Semantic Textual Relatedness. The official evaluation metric for this task is the Spearman's rank correlation coefficient, which captures how well the system-predicted rankings of test instances align with human judgments.

Language	Score (Spearman)	Ranking
Algerian Arabic	0.6035781253	6
Spanish	0.7171198162	7
Moroccan Arabic	0.7893667707	12
English	0.8345843316	13

Table 4: Results of our proposed models on SemEval 2024 Task 1 Track A : Supervised Semantic Textual Relatedness test set

Based on the experimental results, our approaches for SemEval 2024 Task 1 Track A: Supervised Semantic Textual Relatedness demonstrated competitive performance across multiple languages. Here's a summary of our findings:

- **Algerian Arabic** : Our model achieved a score of 0.6035781253, ranking 6th out of 24 submissions. This indicates that our approach effectively captured the semantic relatedness between sentence pairs in Algerian Arabic, outperforming a significant portion of the participating systems.

- **Spanish** : In Spanish, our model achieved a score of 0.7171198162, securing the 7th position out of 25 submissions. This suggests that our approach successfully captured semantic relationships in Spanish text, performing competitively compared to other systems.

- **Moroccan Arabic** : Our model attained a score of 0.7893667707, ranking 12th out of 23 submissions in Moroccan Arabic. While our performance in this language was slightly lower compared to others, our approach still demonstrated notable effectiveness in capturing semantic relatedness in Moroccan Arabic text.

- **English** : For English, our model achieved a score of 0.8345843316, placing 13th out of 36 submissions. Despite the larger number of submissions in English, our approach still showcased strong performance, indicating its capability to accurately assess semantic relatedness in English sentence pairs.

Overall, our experimental results highlight the robustness and effectiveness of our proposed approaches across different languages in capturing semantic textual relatedness. These findings underscore the potential of transformer-based models fine-tuned with appropriate hyperparameters and loss functions to excel in tasks requiring semantic understanding of textual data. Additionally, the competitive rankings across multiple languages signify the versatility and generalizability of our approach, further validating its suitability for real-world applications requiring accurate assessment of semantic relatedness in diverse linguistic contexts.

5 Conclusion

In conclusion, this paper has presented our system and findings for SemEval 2024 Task 1 Track A: Supervised Semantic Textual Relatedness. The primary objective of this task was to detect the degree of semantic relatedness between pairs of sentences across multiple languages. Our submitted models, leveraging various transformer-based architectures including MARBERT-V2, mDeBERTa-V3-Base, DarijaBERT, and DeBERTa-V3-Large, fine-tuned with different loss functions such as Huber Loss, Mean Absolute Error, and Mean Squared Error, achieved competitive rankings across different language tracks.

Our approach highlights the effectiveness of leveraging advanced transformer-based models and fine-tuning techniques to capture intricate semantic relationships within textual data. By incorporating diverse loss functions during the training process, we aimed to optimize the models' performance and enhance their robustness across various linguistic contexts.

Moving forward, further research in semantic textual relatedness should focus on refining existing methodologies, exploring novel architectures, and addressing cross-lingual challenges. Additionally, efforts to incorporate additional linguistic features and develop more comprehensive evaluation metrics can contribute to advancing the state-of-the-art in this field.

Overall, our contributions underscore the significance of semantic textual relatedness in natural language processing tasks and pave the way for the development of more sophisticated and context-aware systems capable of understanding and interpreting textual data with greater precision and accuracy.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Kamel Gaanoun, Abdou Mohamed Naira, Anass Allak, and Imade Benelallam. 2024. Darijabert: a step forward in nlp for the written moroccan dialect. *International Journal of Data Science and Analytics*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

ZXQ at SemEval-2024 Task 7: Fine-tuning GPT-3.5-Turbo for Numerical Reasoning

Zhen Qian, Xiaofei Xu, Xiuzhen Zhang
School of Computing Technologies, RMIT University, Australia
S3888611@student.rmit.edu.au
S3833028@student.rmit.edu.au
xiuzhen.zhang@rmit.edu.au

Abstract

In this paper, we present our system for the SemEval-2024 Task 7, i.e., NumEval subtask 3: Numerical Reasoning. Given a news article and its headline, the numerical reasoning task involves creating a system to compute the intentionally excluded number within the news headline. We propose a fine-tuned GPT-3.5-turbo model, specifically engineered to deduce missing numerals directly from the content of news articles. The model is trained with a human-engineered prompt that integrates the news content and the masked headline, tailoring its accuracy for the designated task. It achieves an accuracy of 0.94 on the test data and secures the second position in the official leaderboard. An examination on the system's inference results reveals its commendable accuracy in identifying correct numerals when they can be directly "copied" from the articles. However, the error rates increase when it comes to some ambiguous operations such as rounding.

1 Introduction

Huang et al. (2023) noted a deficiency in contemporary encoder-decoder models when applied to headline generation, specifically addressing imprecisions in the numerals within the generated headlines. To facilitate a thorough investigation of this issue, the authors introduced a novel dataset (i.e., NumHG dataset[1]), consisting of over 21,000 news articles rich in numerals and accompanied by detailed annotations. The dataset is linked to two sub-tasks. The first sub-task centres on numerical reasoning, requiring models to calculate the missing numerals within the headline based on the news articles. The second sub-task focuses on headline generation, requiring models to generate a headline grounded in the provided news content.

This paper focuses on the first subtask of numeral reasoning, aiming to assess the fine-tuned GPT 3.5 turbo's performance in handling numerical reasoning tasks within the context of the newly introduced dataset. Inspired by the idea of instruction tuning (Wei et al., 2022a), we carefully design the textual prompts for training GPT 3.5 turbo to calculate the missing number in the masked headline. Additionally, drawing from the concept of mapping reasoning problems to annotations alongside final answers (Amini et al., 2019; Chiang and Chen, 2019), we carry out experiments to utilize the annotations given in the NumHG dataset. The best fine-tuned model from our experiments achieves an accuracy of 0.94, securing the second position on the official leaderboard. However, we acknowledge that our best model does not rely on the annotations provided in the dataset to generate numerals. Instead, the best model calculates numerals based solely on the news content.

2 Related Work

This work draws inspiration from two key research areas: instruction tuning and leveraging intermediate reasoning steps for solving math word problems. Instruction tuning shows that incorporating prompts or instructions into training datasets, as proposed by Wei et al. (2022a), can improve the language models' performance on unseen data. Sanh et al. (2022) also suggest that using diverse prompts to augment training datasets can help language models to achieve better generalization. Regarding the use of intermediate reasoning steps, some researchers express these steps in symbolic format, such as Chiang and Chen (2019) who train language models to generate equations leading to final answers, and Amini et al. (2019) who map math word problems to predefined operations. Others opt for natural language descriptions, like Cobbe et al. (2021), who propose a system utilizing a language model to generate reasoning steps and final answers in natural language, along with a verifier

[1] https://github.com/ArrowHuang/NumHG.git

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 218–223

Operator	Description
Copy	Copy from the article
Trans	Convert into a number
Paraphrase	Paraphrase the form of digits to other representations
Round	Hold some digits after the decimal point of a given numeral
Subtract	Subtract a from b
Add	Add a and b
Span	Select a span from the article
Divide	Divide a by b
Multiply	Multiply a and b

Table 1: overview of the pre-defined operations given in the dataset.

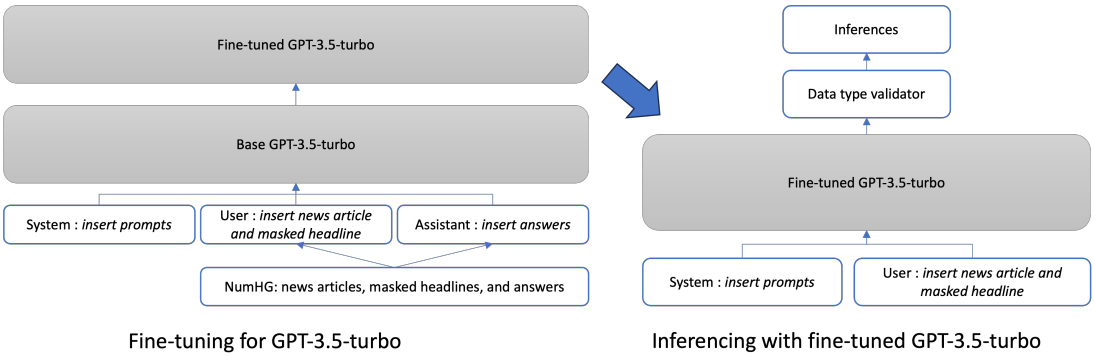


Figure 1: fine-tuning and deploying process for the GPT-3.5-turbo

model to assess the generated answers.

3 Problem Definition

The NumHG dataset comprises 21,157 pieces of news articles, each ranging between 200 to 300 words in the content. The headlines of these news articles include one or more numerals, with one numeral intentionally omitted from each headline. The novelty of the NumHG dataset lies in its provision of meticulous annotations detailing the computational processes used to arrive at the missing numerals in the headlines. This dataset articulates 12 fundamental operations for numeral computation, encompassing actions such as copying, addition, and multiplication. The pre-defined operations are listed in Table 1 (Huang et al., 2023). While in certain cases, computing the correct numerals may require a single operation, such as straightforwardly copying the numerals from the news content as the correct answers, in other instances, it may involve a sequential combination of multiple operations. For example, this could involve rounding the result after adding two numerals found in the news content. The sub-task of numerical reasoning requires us to develop a system capable of accurately calculating the omitted numerals based on the news contents and potentially utilizing the provided annotations.

4 System Overview

In this section, we provide a detailed introduction to our system. Figure 1 illustrates a general outline of our system.

4.1 Model Description

This paper aims to assess the performance of a fine-tuned GPT-3.5-turbo-0613 model in numerical reasoning, specifically its accuracy in computing the desired numerals within news headlines based on content of the news articles. Through the process of fine-tuning, we aim to harness the capability of the GPT-3.5-turbo model by training it with the NumHG dataset, tailoring its performance to our designated task. As of the period during which our experiments were conducted, GPT-3.5-turbo-0613, unveiled in June 2023, stands as the most recent iteration released by OpenAI. This version offers enhanced steerability and reduced costs associated with input tokens. The experiments conducted centre around the refinement of the base model through the fine-tuning process.

The fine-tuned model is then employed for making inferences on the test data. We also imple-

role	content
system	“you will be given a piece of news with prefix ’news:’. you will also be given an incomplete headline with the prefix ’masked headline:’. based on the news content, please output the missing number in the masked headline. please ensure the output is the number only rather than the whole sentence.”
user	“news: news content for each instance in the dataset. masked headline: masked headline for each news article in the dataset”
assistant	“the omitted number from the headline”

Table 2: prompts employed in experiment 1 – training the model to calculate the numerals directly from the news content

role	content
system	“you will be given a piece of news with prefix ’news:’. you will also be given an incomplete headline with the prefix ’masked headline:’. based on the news content, please output how the missing number in the masked headline is calculated together with the final answer. please ensure the operations follow the format below: Copy(v), Trans(e), Paraphrase(v, n), Round(v, c), Subtract(v0, v1), Add(v0, v1), Span(s), Divide(v0,v1), Multiply(v0,v1).”
user	“news: news content for each instance in the dataset. masked headline: masked headline for each news article in the dataset”
assistant	"calculations";“the omitted number”

Table 3: prompts employed in experiment 2 – training the model also to generate the operations

ment a program for verifying the data types of the model’s outputs. For the sub-task of numerical reasoning, the expected outputs should be numerical values. Our program systematically converts any non-numerical outputs to the value of 0.

4.2 Prompt Design

To fine-tune a GPT-3.5-turbo using the OpenAI API, it is necessary to convert each instance in the dataset into a format compatible with the model[2]. This involves defining the content for three distinct roles for each instance. The roles include system, user, and assistant. The content assigned to the system role consists of instructions and prompts directed towards the model. For the user role, the corresponding content involves inputs provided to the model, including questions. The content allocated to the assistant role consists of the expected outputs or answers. We mainly carried out two experiments. For the first experiment, we train the model to calculate the numerals directly from the news content. For the second, in addition to the numerals, we also instruct the model to generate the operations required to reach the numerals. The specific prompts employed in this process are presented in Table 2 and Table 3. More detailed examples are shown in Figure 2 and Figure 3.

[2] https://platform.openai.com/docs/guides/fine-tuning

5 Experiment setup

The dataset allocated for the sub-task of numerical reasoning comprises 21,157 instances, which is further split into 80% for training data and 20% for test data. Fine-tuning of the GPT-3.5-turbo-0613 model is performed via the OpenAI API, adhering to the guidelines outlined in the OpenAI API documentation. The hyper-parameters for model training, including learning rate, batch size, and epochs, are configured to auto. Throughout the training phase, evaluation metrics such as training loss, training token accuracy, validation loss, and validation token accuracy are provided by OpenAI. Following the completion of the training process, the fine-tuned model is employed to generate inferences on the test data with temperature set to default.

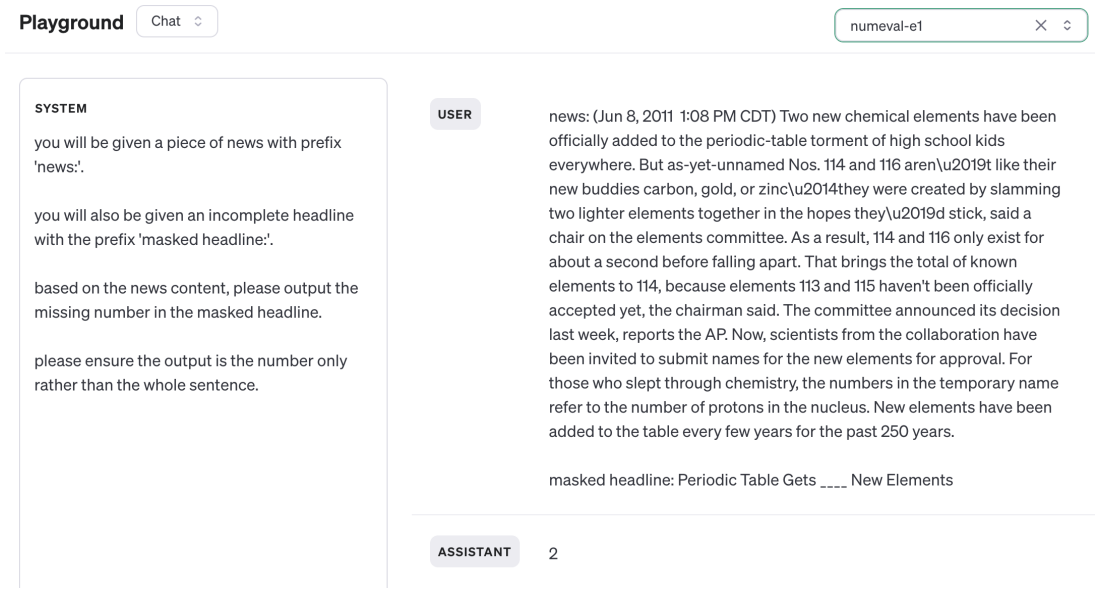


Figure 2: one example for the prompts used for instructing the model to calculate the numerals directly

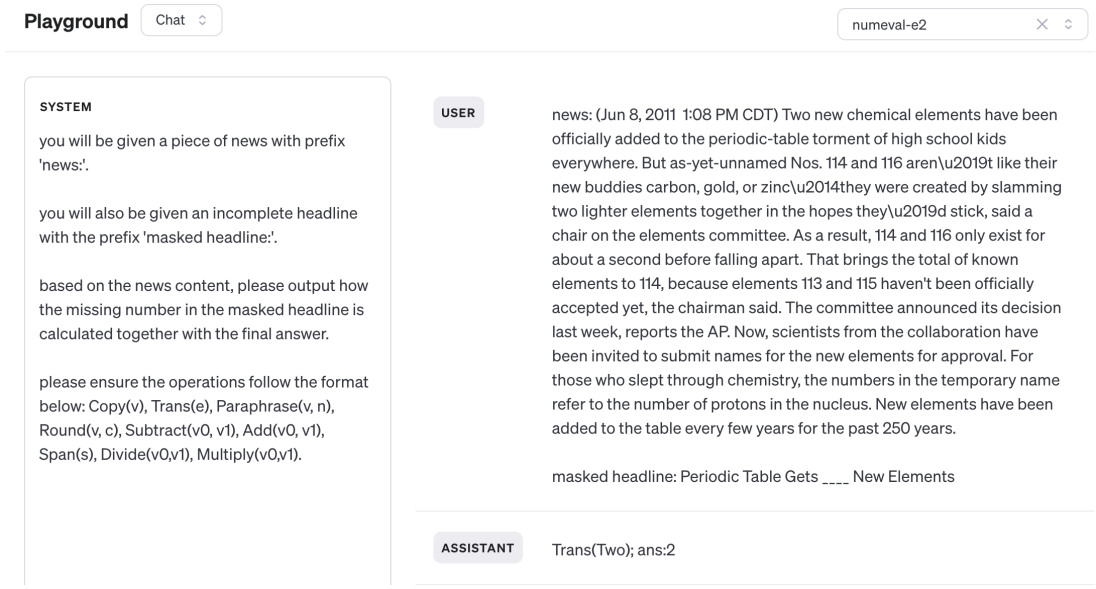


Figure 3: one example for the prompts used for instructing the model to also generate the operations

6 Results

The fine-tuned GPT-3.5-turbo-0613 model from the experiment 1, which calculates the numerals directly from news content, demonstrates a commendable accuracy of 0.94, securing the second position on the leaderboard. The model from the experiment 2, which also output the intermediate operations required to arrive at the numerals, only achieves an accuracy of 0.90. An analysis of model errors has been conducted, with detailed statistics presented in Table 4.

The test data comprises 4,921 instances, featuring a total of 5,237 operators in the annotations. Table 4 reveals that error rates, for both models, are notably low for operations such as Copy, Trans, and Paraphrase, while they are comparatively high for Round, Multiply, Add, and Divide. Given that 88.5% of the operations in the test data pertain to Copy, Trans, and Paraphrase, the model's commendable performance in these three operations significantly contributes to its overall accuracy.

It is important to note that our models cannot detect unanswerable questions in the test data. This anomaly arises from the fact that there are no unanswerable questions in the training data. The models do not learn to predict unanswerable questions in

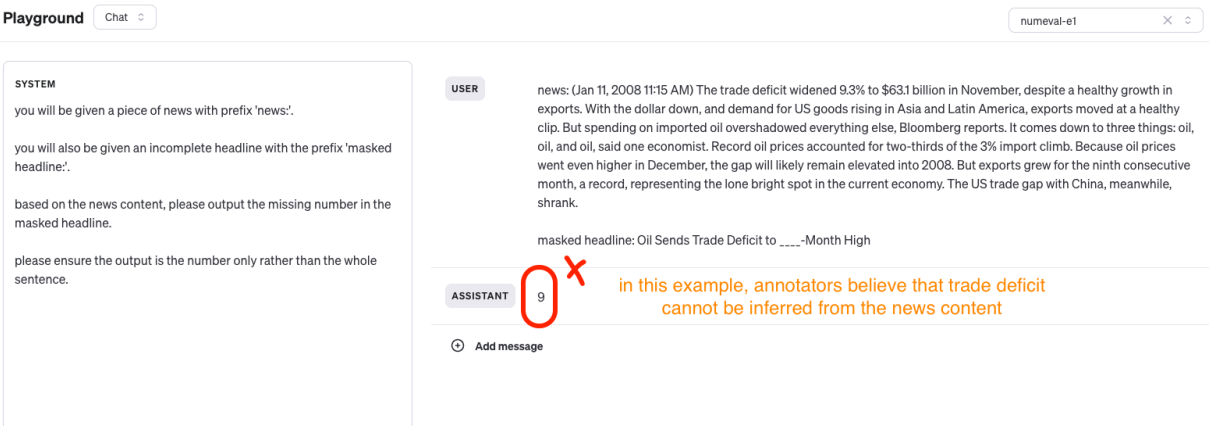


Figure 4: one example for unanswerable questions

Operator	Model 1 Error Rate	Model 2 Error Rate
Copy	4.32%	6.79%
Trans	3.13%	11.37%
Paraphrase	7.39%	15.43%
Round	41.08%	62.70%
Subtract	18.69%	43.93%
Add	24.00%	49.00%
Span	10.34%	30.17%
Divide	23.68%	55.26%
Multiply	35.71%	52.38%

Table 4: error analysis on the test data

the training phase. The unanswerable questions refer to instances identified by annotators when the numerals cannot be inferred from the news content. Figure 4 shows one example of the unanswerable questions in the test data. Another anomaly arises as the model trained for generating intermediate operations shows lower accuracy, contradicting prior works' conclusion that incorporating intermediate reasoning steps in symbolic formats should enhance language model performance by Chiang and Chen (2019) and Amini et al. (2019). To improve the model's ability to utilize annotations and boost overall accuracy, future research should explore alternative methods such as experimenting with different language models, adjusting model architecture, and employing Chain-of-Thought prompting (Wei et al., 2023; Ling et al., 2023).

7 Conclusion

In this paper, we propose to finetune the GPT-3.5-turbo specifically tailored for handling numerical reasoning in the headline generation context. Through the carefully engineered prompt that aggregates the content of news articles and the masked headlines during the fine-tuning process, we achieved an accuracy of 0.94 and ranked second in the official leaderboard. However, our model exhibits relatively high error rates particularly in operations such as rounding numbers. Additionally, further development is needed to better utilize the annotations to improve the model's accuracy.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367. Association for Computational Linguistics.

Ting-Rui Chiang and Yun-Nung Chen. 2019. Semantically-aligned equation generation for solving and reasoning math word problems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2656–2668. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. arXiv:2309.01455.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. 36:36407–36433.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

BAMO at SemEval-2024 Task 9: BRAINTEASER: A Novel Task Defying Common Sense

Baktash Ansari, Mohammadmostafa Rostamkhani, Sauleh Eetemadi
Iran University of Science and Technology
{baktash_ansari, mo_rostamkhani97}@comp.iust.ac.ir, sauleh@iust.ac.ir

Abstract

This paper outlines our approach to SemEval 2024 Task 9, BRAINTEASER: A Novel Task Defying Common Sense. The task aims to evaluate the ability of language models to think creatively. The dataset comprises multi-choice questions that challenge models to think 'outside of the box'. We fine-tune 2 models, BERT and RoBERTa Large. Next, we employ a Chain of Thought (CoT) zero-shot prompting approach with 6 large language models, such as GPT-3.5, Mixtral, and Llama2. Finally, we utilize ReConcile, a technique that employs a 'round table conference' approach with multiple agents for zero-shot learning, to generate consensus answers among 3 selected language models. Our best method achieves an overall accuracy of 85 percent on the sentence puzzles subtask.

1 Introduction

Evaluation methods in the NLP community predominantly emphasize Vertical thinking, characterized by sequential, analytical processes based on rationality, logic, and rules. However, SemEval-2024 Task 9, BRAINTEASER (Jiang et al., 2024b), which is based on the original BRAINTEASER dataset (Jiang et al., 2023), aims to introduce a task that promotes lateral thinking (or "thinking outside the box"), a divergent and creative process involving the exploration of new perspectives when addressing problems. The BRAINTEASER QA task consists of two subtasks for the English language: Sentence Puzzles and Word Puzzles. This task is designed to challenge the common sense reasoning capabilities of NLP models and stimulate the development of models that can think laterally.

- **Sentence Puzzles**: Sentence-type brain teaser where the puzzle-defying commonsense is centered on sentence snippets.
- **Word Puzzles**: Word-type brain teaser where the answer violates the default meaning of the word and focuses on the letter composition of the target question.

We generate baselines for two attention-based models, BERT (Devlin et al., 2019) and RoBERTa-Large (Liu et al., 2019), as selected in the task paper, to solve these types of multiple-choice problems. Then we fine-tune them with the same configs. After achieving some accuracy through fine-tuning, we explore zero-shot prompting with various large language models (LLMs). To further improve results, zero-shot prompting is conducted using a Chain of Thought technique (Wei et al., 2023). As illustrated in Figure 1, we compel the model to analyze and provide step-by-step reasoning for its answer instead of simply providing a correct option alone. This approach helps the model focus more on details and answer questions with fewer errors.

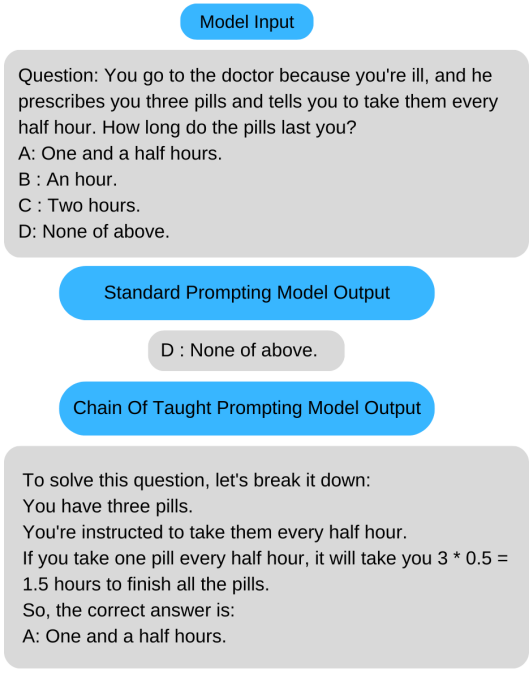


Figure 1: Chain Of Thought Prompting (GPT3.5)

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 224–232

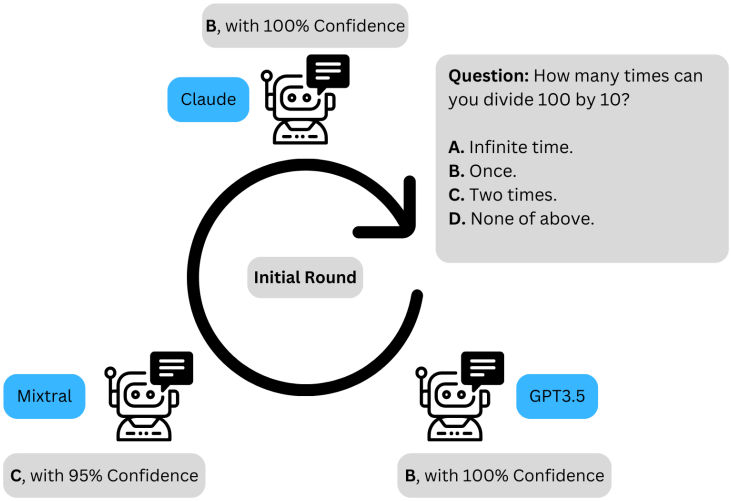


Figure 2: An Illustration of RECONCILE for Initial Round

Communication among multiple agents is fundamentally important in complex decision-making processes. Therefore, as our main strategy, we use the ReConcile technique (Chen et al., 2023), which generates answers by gathering the consensus of multiple models based on their confidence levels, to address these problems. To make this technique compatible with this type of dataset, we extend the application of ReConcile from yes/no questions to the BRAINTEASER questions. In this system, we apply the concept of a society of minds to multiple agents. For round table decision-making, we execute three phases: Initial Response Generation, Multi-Round Discussion, and Final Answer Generation on three language learning models: Mixtral8x7b (Jiang et al., 2024a), Claude [1], and GPT3.5 [2]. As illustrated in Figure 2, in each phase, we generate specific prompts (see Appendix B) for models to answer the question, along with their confidence level between 0 and 1. We then use their answers for the next round and derive an overall consensus answer for each round. The method that gave us our best result achieved a rank of 11 out of 33 for the sentence puzzles in the task leaderboard. Further details of our implementations are available through our GitHub repository. [3].

2 Background

2.1 Related Works

The exploration of reasoning abilities in large language models, lateral thinking, and common sense reasoning has been the focus of several studies in recent years. The BRAINTEASER is a novel task in this context, requiring a unique blend of these capabilities. In the paper (Zhang et al., 2022), foundational insights into the use of knowledge graphs for self-supervision in common sense reasoning tasks are provided. This work is particularly relevant as it demonstrates how external knowledge can significantly improve an LLM's ability to solve reasoning problems. Furthering the discussion on reasoning, LatEval (Huang et al., 2024) introduces an evaluation benchmark for LLMs based on lateral thinking puzzles. This process is similar to the method in solving the BRAINTEASER, which encourages models to elaborate on their reasoning, leading to improved performance. The paper RiddleSense (Lin et al., 2021) offers an analysis of how LLMs handle riddle questions that require linguistic creativity and common sense knowledge. This is directly linked to our approaches, where the ability of an LLM to use analogous examples influences its problem-solving effectiveness. Also, (Dou and Peng, 2022) investigates the inherent knowledge within language models and their ability to function in zero-shot common sense question answering tasks. The integration of external knowledge is particularly relevant to tasks requiring common sense and real-world information, underscoring the importance of knowledge retrieval in the context of the BRAINTEASER. MVP-Tuning (Huang et al., 2023) introduces a novel approach to knowledge retrieval using prompt tuning. This aligns with our aim in solving BRAINTEASER to evaluate the capacity of LLMs to handle questions that necessitate an understanding of the world as humans

[1] Available at https://claude.ai/ .
[2] Available at https://openai.com/ .
[3] Github Repository

perceive it. Lastly, ReConcile (Chen et al., 2023) and (Liang et al., 2023) both discuss the use of multiple LLMs to improve reasoning capabilities. These works highlight the potential of using a diverse set of models to solve complex tasks like the BRAINTEASER, further enriching the strategies for solving such tasks with AI systems.

2.2 Datasets

The organizers provide datasets for one language: English. As mentioned previously, the dataset consists of two categories: Sentence Puzzles and Word Puzzles. The task providers construct reconstruction versions of the original data in two parallel ways: Semantic Reconstruction and Context Reconstruction. This is done to ensure that the task evaluates lateral thinking ability rather than mere memorization. The Semantic variant reformulates the initial question while preserving its answer, whereas the Context variant retains the misleading commonsense assumption as is and modifies the question and its answer to fit a different situational context.

The dataset is split into two parts for the evaluation phase: train and test sets for each category. In the sentence puzzles category, the train set comprises 507-row samples, while the test set consists of 120-row samples. Similarly, in the word puzzles category, the train set contains 396-row samples, and the test set has 96-row samples. Each sample includes a question with its corresponding answer and three distractors. We utilize both word and sentence puzzle datasets during the training phase, but only the sentence puzzles dataset is used for zero-shot phases.

2.3 Evaluation Metrics

The accuracy metric is employed for evaluation as described in the task paper. Performance evaluation is conducted using two accuracy metrics: Instance-based Accuracy and Group-based Accuracy for Original, Semantic, and Context questions.

3 System overview

3.1 Preprocessing

In the fine-tuning phase, we employ two transformer-based models, BERT and RoBERTa, for multiple choice tasks. Both models are pre-trained on large text corpora. The input to these models is a sequence constructed by concatenating the question with each choice, separated by special tokens. This process is facilitated by the models' tokenizers, which convert the text into a format that the models can understand. For a given question "Q", and choices, the input to the models would be:

$$\text{Input}_i = \text{[CLS] Q [SEP]Choice}_i\text{[SEP]} \quad (1)$$

where "i" represents the index of the choice. Each sequence represents a different choice, and the models' understanding of the context and the choices allows them to predict the correct answer. The fine-tuning process adapts the models to this specific task, optimizing their parameters to minimize the difference between the predicted and actual answers.

3.2 Model Training

We utilize BERT-Base and RoBERTa-Large models, along with their respective tokenizers, for word embeddings in the multiple choice task, sourced from the Hugging Face library. These models are finetuned using the Hugging Face trainer. Initially, we load the models and establish a baseline on the test set for both sentence and word puzzles. Subsequently, each model undergoes finetuning with the default Hugging Face Cross-Entropy loss function for classification, and overall accuracy is computed as the metric using the same hyper-parameter configurations as detailed in Table 1.

Hyperparameter	Value
Learning Rate	1×10^{-5}
Optimizer	Adam
β_1, β_2	$0.9, 0.999$
Weight Decay	0.01
Batch Size	1
Loss Function	Cross-Entropy
Logging Steps	100
Evaluation Metric	Accuracy
Global Seed	255

Table 1: Hyperparameters Configuration

3.3 Chain of Thought Prompting

We use 6 LLMs for CoT zero-shot: Mixtral(8x7b), Claude, GPT3.5, Llama-2-70b (Touvron et al., 2023), OpenChat (Wang et al., 2023), and Microsoft Copilot [4]. We use graphical web page interfaces for Claude, GPT3.5(chatGPT), and Microsoft

[4]Available at https://copilot.microsoft.com/

Copilot (precise mode). We use the same prompt as the ReConcile initial round. Microsoft Copilot gives the best performance of this section.

3.4 ReConcile Round Table

Models make mistakes in one or more types of questions and cannot provide the correct answer on the first attempt. We need to ask them to pay attention to certain parts of the question or give hints to the model so it can provide the correct answer. To ensure human involvement is minimized and models can help each other, we have employed the ReConcile method. Using this approach, each model complements the other. The process of this system is as follows:

- **Initial Response Generation**: First, using an initial prompt, we ask each model to provide the answer to the question, provide a reason for the answer, and declare a confidence level between 0 and 1.
- **Multi-Round Discussion**: We give the responses, reasoning, and confidence levels of the three models, along with the initial prompt, as input to the models once again. This enables them to consider both the context of the question and the responses of the three models when making a selection.
- **Final Answer Generation.** In this stage, we initialize a weight for each of the 4 options of the question, and these weights are summed up with the confidence level of each model. Finally, the option with the highest weight is chosen as the correct option.

Let's denote the confidence of the model m_i for its selected choice c_j as $conf(m_i, c_j)$. Then, the total confidence of each choice can be calculated as:

$$\text{TC}(c_j) = \sum_{i=1}^{3} \text{conf}(m_i, c_j) \tag{2}$$

Where the sum is over all models that selected choice c_j. Finally, the choice with the highest total confidence is selected as the correct choice:

$$\text{c}_{\text{correct}} = \arg\max_{c_j \in C} TC(c_j) \tag{3}$$

This means that the correct choice is the one that maximizes the total confidence over all choices.

The notable point is that this method should be implemented by models that roughly have equal performance to grow together after several rounds. If a model has much lower performance compared to other models, its reasoning and confidence level may negatively affect others. For this reason, we performed this task on three models: Mixtral8x7b, GPT3.5, and Claude, which have almost similar accuracy in the initial round. This iterative process can be continued until all models reach a consensus and all agree on a specific option for the questions. We repeated this process for two discussion rounds.

4 Experiments and Results

4.1 Experimental Setup

The training and test sets of the sentence and word puzzle datasets are used with a split of 0.8 and 0.2, respectively for fine-tuning. Additionally, we utilize Google Colab's T4 GPU with the hyperparameters as shown in Table 1. For zero-shot prompting, we use the 120-row test set from the sentence puzzles.

We leverage HuggingChat [5] for Mixtral8x7b, OpenChat, and Llama2-70b. Furthermore, we utilize the official web interface of Claude, Microsoft Copilot, and GPT3.5. Results for each model and their corresponding training codes are available in the GitHub repository.

4.2 Results

As illustrated in Table 2 for the sentence puzzles and Table 3 for the word puzzles, we present the performance of BERT and RoBERTa Large in both their base and fine-tuned versions.[6] We load the best model based on Overall Accuracy at the end of each training. The best performance is achieved by RoBERTa for both sentence and word puzzles. As illustrated in Appendix A For every 100 training steps, we log the overall accuracy for two models.

The submission scores computed by the task organizer for CoT zero-shot are available in Table 4. Among these LLMs, Microsoft Copilot achieves the best performance. The success of LLMs in responding to these questions depends on the model's ability to recognize that these questions are tricky and that it doesn't need to provide logical reasoning in many cases; the question merely plays with words. Microsoft Copilot understood this phenomenon in many questions. However, it

[5] Available at https://huggingface.co/chat/

[6] S(sentence), ori(original), sem(semantic), con(context)

Model	Type	S_ori	S_sem	S_con	S_ori_sem	S_ori_sem_con	S_overall
BERT Base	Baseline	0.400	0.450	0.325	0.350	0.175	0.391
RoBERTa Large	Baseline	0.250	0.175	0.275	0.175	0.050	0.233
BERT Base	Finetune	0.725	0.750	0.650	0.725	0.575	0.708
RoBERTa Large	Finetune	**0.800**	**0.775**	**0.725**	**0.775**	**0.700**	**0.766**

Table 2: Models' Performance on Sentence Puzzles

Model	Type	W_ori	W_sem	W_con	W_ori_sem	W_ori_sem_con	W_overall
BERT Base	Baseline	0.562	0.343	0.375	0.281	0.093	0.427
RoBERTa Large	Baseline	0.250	0.281	0.343	0.218	0.093	0.291
BERT Base	Finetune	**0.687**	0.656	0.468	0.625	0.375	0.604
RoBERTa Large	Finetune	**0.687**	**0.687**	**0.562**	**0.656**	**0.468**	**0.645**

Table 3: Models' Performance on Word Puzzles

also made mistakes in several questions. For example, consider Appendix C for an illustration.

For Reconcile, the results of each model in every round, as well as the consensus reached in each round, are presented in Table 5. In the table, we observe that Claude achieves the highest overall accuracy among the models in each round. Nearly every model in the Reconcile system either improves or maintains its best performance in overall accuracy with each round. This suggests that they are all capable of making informed decisions based on the reasoning provided by all agents during the discussion rounds. At the conclusion of round 2, the consensus overall accuracy stands at 0.758, which is 0.3 to 0.5 points higher than the initial round results of all three models. Furthermore, we note that the consensus result improves by approximately 1 percent from the initial round to round 1, and by approximately 0.8 percent from round 1 to round 2. This indicates that after several rounds, the models converge and reach a consensus on the questions. See Appendix D. Also for an example, see Appendix E

5 Conclusion

In this paper, we present our approach to SemEval 2024 Task 9, BRAINTEASER: A Novel Task Defying Common Sense, which challenges models to think creatively beyond conventional reasoning. Through fine-tuning BERT and RoBERTa models, as well as employing zero-shot prompting techniques using various large language models, we achieved notable performance improvements. Particularly, Microsoft Copilot performs the best without being taught beforehand, showing it understands the tricky task really well. Furthermore, our ReConcile Round Table method demonstrates the efficacy of collaborative decision-making among models, leading to a progressive improvement in overall accuracy across multiple rounds of discussion.

References

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023. Reconcile: Round-table conference improves reasoning via consensus among diverse llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou and Nanyun Peng. 2022. Zero-shot commonsense question answering with cloze translation and consistency optimization.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, and Hai-Tao Zheng. 2024. Lateval: An interactive llms evaluation benchmark with incomplete information from lateral thinking puzzles.

Yongfeng Huang, Yanyang Li, Yichong Xu, Lin Zhang, Ruyi Gan, Jiaxing Zhang, and Liwei Wang. 2023. MVP-tuning: Multi-view knowledge retrieval with prompt tuning for commonsense reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13417–13432, Toronto, Canada. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris

Model	S_ori	S_sem	S_con	S_ori_sem	S_ori_sem_con	S_overall
Open Chat	0.500	0.500	0.525	0.375	0.300	0.508
Llama-2-70b	0.600	0.625	0.550	0.550	0.400	0.591
Mixtral-8x7b	0.750	0.700	0.650	0.650	0.525	0.700
GPT3.5	0.750	**0.775**	0.625	0.650	0.500	0.710
Claud	**0.775**	0.725	0.700	**0.725**	0.625	**0.730**
Microsoft Copilot	**0.925**	**0.900**	**0.775**	**0.875**	**0.750**	**0.860**

Table 4: LLM's Zero-Shot Performance on Sentence Puzzles

Model	S_ori	S_sem	S_con	S_ori_sem	S_ori_sem_con	S_overall
Initial Round						
GPT3.5	0.750	**0.775**	0.625	0.650	0.500	0.710
Claude	**0.775**	0.725	**0.700**	**0.725**	0.625	**0.730**
Mixtral-8x7b	0.750	0.700	0.650	0.650	0.525	0.700
Consensus	**0.775**	**0.750**	**0.700**	**0.675**	0.575	0.740
Round 1						
GPT3.5	0.775	0.725	0.675	0.700	0.570	0.720
Claude	**0.800**	**0.775**	**0.725**	**0.725**	**0.600**	**0.760**
Mixtral-8x7b	0.700	0.725	0.625	0.625	0.450	0.680
Consensus	**0.800**	**0.750**	**0.700**	**0.725**	**0.600**	**0.750**
Round 2						
GPT3.5	**0.800**	**0.800**	0.675	**0.750**	**0.600**	**0.750**
Claude	**0.800**	0.775	0.725	0.725	**0.600**	**0.760**
Mixtral-8x7b	0.725	0.725	**0.725**	0.675	0.550	0.725
Consensus	**0.775**	**0.800**	**0.700**	**0.725**	**0.600**	**0.758**

Table 5: Reconcile Results

Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024b. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Moly-

bog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Jiarui Zhang, Filip Ilievski, Kaixin Ma, Jonathan Francis, and Alessandro Oltramari. 2022. An empirical investigation of commonsense self-supervision with knowledge graphs.

A Training logs

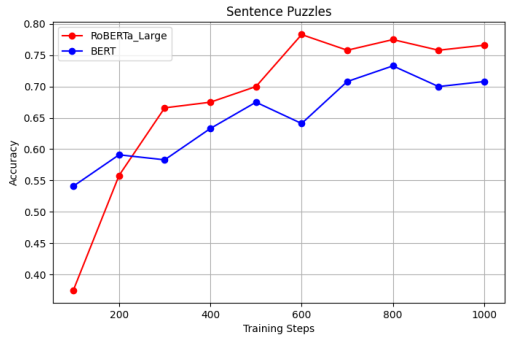


Figure 3: Overall Accuracy of Two Models Logged Every 100 Training Steps on Sentence Puzzles.

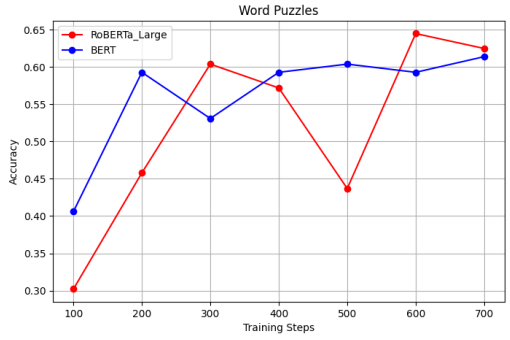


Figure 4: Overall Accuracy of Two Models Logged Every 100 Training Steps on Word Puzzles.

B Prompts

Initial Prompt

Please answer the question with step-by-step reasoning. Also, evaluate your confidence level (between 0.0 and 1.0) to indicate the possibility of your answer being right. Please give the answer in this template:
Solution:
Reasoning:
{**Question**}
{**Choices**}

Discussion Prompt

{**initial_prompt**}
Carefully review the following solutions from other agents as additional information, and provide your own answer and step-by-step reasoning to the question. Clearly state which point of view you agree or disagree with and why.
{**agent_one_answer**}
{**agent_one_reason**}
{**agent_one_confidence**}
{**agent_two_answer**}
{**agent_two_reason**}
{**agent_two_confidence**}
{**agent_three_answer**}
{**agent_three_reason**}
{**agent_three_confidence**}

Figure 5: ReConcile Initial and Discussion Prompts

C Copilot Examples

As illustrated in Table 6, the model correctly answers the first three samples but provides an incorrect response for the fourth. This model lacks accuracy in calculation questions, often correctly presenting the equation but solving it inaccurately. Additionally, in some cases, the model accurately formulates the equation and even solves it correctly. However, when the desired option is not among the choices, instead of choosing the 'None of the above' option, it attempts to reinterpret the question in a way that one of the numerical options becomes correct.

D Reconcile Convergence

Table 7 indicates the number of questions to which all three models provide the same answer choice in each round. This suggests that after each round, the models achieve a relative convergence.

Question	Answer
How can you toss a fragile glass ball against a sturdy brick wall from a distance without shattering it? A. No problem, because sturdy bricks are very hard to crack B. You need to throw the glass ball hard. C. You need to throw the glass ball gently. D. None of above.	Solution: A: No problem, because sturdy bricks are very hard to crack Reasoning: The question is a play on words. It's not the glass ball that we're worried about cracking, it's the brick wall. And brick walls are indeed very hard to crack.
Ana's school has 4 students in Alaska: Maria, James, Jonghyun. What is the name of other student? A. Mark. B. Ana C. Michael D. None of above.	Solution: B: Ana. Reasoning: The question states "Ana's school has 4 students in Alaska: Maria, James, Jonghyun." So, Ana is the fourth student.
In a house, there are three major rooms. The first is stuffed with cash. The second is stuffed with crucial documents. The third is stuffed with jewels. One day, all of these rooms caught fire. Which room did the cops initially put out the fire? A. The one with jewelry. B. The one with important papers. C. The one filled with money. D. None of above.	Solution: D: None of above. Reasoning: The question is a play on words. It's not the cops' job to put out fires, it's the firefighters' job.
During a 5-day competition, a professional fisherman caught 30 fish. He caught three more fish each day than the day before. On the first day, how many fish did the fisherman catch? A. Two B. One C. Three D. None of above	Solution: A: Two. Reasoning: If the fisherman caught 30 fish in 5 days, and each day he caught three more fish than the day before, we can set up the equation: x + (x+3) + (x+6) + (x+9) + (x+12) = 30. Solving for x gives us x = 2. So, the fisherman caught two fish on the first day.

Table 6: Microsoft Copilot Answers to Some Questions

Round	Question Count
Initial Round	74
Round 1	88
Round 2	104

Table 7: ReConcile Models Convergence

E Reconcile Example

The iterative process of answering the question below is illustrated in Table 8.

If a boat is parked in a marina and tied to a nearby station, how did its owner found the boat a couple of hours later in the middle of the ocean?

A. The station wasn't fixed in the marina

B. The dolphins untied the knots.

C. The waves were powerful, which pushed the boat and the marina together.

D. None of above.

Model	Solution	Reasoning	Confidence
Initial Round			
GPT3.5 Solution	A	If the station (presumably a mooring or dock) wasn't securely fixed in the marina, it could have drifted into the ocean, taking the boat with it.	1.0
Claude Solution	B	This explains how the boat ended up drifting into the ocean after being tied up.	0.7
Mixtral Solution	A	If the station wasn't fixed properly in the marina, it's possible that the boat became loose and drifted away into the ocean due to wind or tidal influences. Over time, the gap between the boat and the dock could increase, eventually pushing the boat further out into deeper waters.	0.85
Round 1			
GPT3.5 Solution	A	If the station to which the boat was tied was not properly fixed in the marina, it could have drifted away into the ocean, taking the boat with it.	0.9
Claude Solution	A	I agree with Agents two and three that an unsecured floating station drifting away best explains how the boat ended up adrift in the ocean.	1.0
Mixtral Solution	A	While all options are plausible scenarios for how the boat might have ended up in the ocean, option A provides the most likely explanation given our current understanding of physics and marine environments. Both agent two and agent three suggest that if the station was not fixed properly, it could have been moved by external forces such as wind or tide, thereby pulling the boat along with it. In contrast, option C assumes that the waves were strong enough to move both the marina and the boat simultaneously, an event that seems less probable than the movement of a single object like the station. Option B attributes the boat's displacement to intelligent behavior on part of dolphins, which lack the necessary appendages to manipulate rope knots underwater. Therefore, we can infer that the station must have been unsecured, allowing it to drift out into the ocean while carrying the boat with it.	0.95
Round 2			
GPT3.5 Solution	A	If the station to which the boat was tied was not properly fixed in the marina, it could have drifted away into the ocean, taking the boat with it.	0.95
Claude Solution	A	I agree with Agents two and three that an unsecured floating station drifting away provides the most logical explanation for how the boat ended up adrift.	1.0
Mixtral Solution	A	All three agents came to the conclusion that option A is the most likely scenario because it's more feasible that an unsecured floating station would be moved by external forces such as wind or tide, rather than assuming that powerful waves managed to push the marina and the boat together at once, or attributing the boat's displacement to intelligent behavior from dolphins who wouldn't possess the physical capability to manipulate rope knots underwater.	0.98

Table 8: Reconcile Example

yangqi at SemEval-2024 Task 9: Simulate Human Thinking by Large Language Model for Lateral Thinking Challenges

Qi Yang, Jingjie Zeng, Liang Yang*, Hongfei Lin
School of Computer Science and Technology, Dalian University of Technology, China
{2665643739, jjtail}@mail.dlut.edu.cn, {liang, hflin}@dlut.edu.cn

Abstract

This paper describes our system used in the SemEval-2024 Task 9 on two sub-tasks, BRAINTEASER: A Novel Task Defying Common Sense. In this work, we developed a system SHTL, which means simulate human thinking capabilities by Large Language Model (LLM). Our approach bifurcates into two main components: Common Sense Reasoning and Rationalize Defying Common Sense. To mitigate the hallucinations of LLM, we implemented a strategy that combines Retrieval-augmented Generation (RAG) with the the Self-Adaptive In-Context Learning (SAICL), thereby sufficiently leveraging the powerful language ability of LLM. The effectiveness of our method has been validated by its performance on the test set, with an average performance on two subtasks that is 30.1 higher than ChatGPT setting zero-shot and only 0.8 lower than that of humans.

1 Introduction

Human reasoning processes comprise two types of thinking: vertical and lateral. Vertical thinking, also known as linear, convergent, or logical thinking, is a sequential analytical process that is based on rationality, logic, and rules. Meanwhile, lateral thinking is a divergent and creative process that involves looking at a problem from a new perspective and defying preconceptions.The success of language models has inspired the natural language processing community to attend to tasks that require implicit and complex reasoning, relying on human-like Common Sense mechanisms. While such vertical thinking tasks have been relatively popular, lateral thinking puzzles have received little attention. Recently, the team led by Yifan Jiang proposed Task 9 for SemEval-2024, named "BRAINTEASER: A Novel Task Defying Common Sense," (Jiang et al., 2023)(Jiang et al., 2024) aimed at addressing this gap, it was a task on a pure English dataset, testing models' ability to demonstrate lateral thinking and challenge default common sense associations. This shared task explores methods to improve models' lateral thinking capabilities.

In this paper, we introduce our entries into two BRAINTEASE subtasks. Inspired by recent research on using LLM to design Agents (Xi et al., 2023), our approach leverages an LLM to architect a system that adeptly simulates the intricacies of human divergent thinking processes. Specifically, our model capitalizes on the advanced linguistic capabilities inherent within the LLM, thereby obviating the need for supplementary training protocols. This strategy enables our system to demonstrate commendable performance across both targeted subtasks.

Furthermore, we also focus on the issue of hallucinations in LLM. LLM can sometimes generate erroneous or highly inaccurate responses. The tendency for LLM to produce these hallucinations becomes particularly noticeable when confronted with such phenomena in our investigations. To solve this problem, we draw inspiration from RAG (Lewis et al., 2020) strategies. We also discover that the performance of LLM can vary greatly depending on the specific prompt words used. As a result, we develop several sets of prompt words and ultimately select the set that achieved the highest performance on our validation tests.

Our method achieves competitive results on SemEval-2024 task 9, ranking well on all tasks, especially on more fine-grained classification tasks. Our method far exceeds the performance of ChatGPT, and on the officially provided test set, there is only a slight gap with the results of human evaluation.

*Corresponding author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 233–238

Question	Choice
A man shaves everyday, yet keeps his beard long.	He is a barber. He wants to maintain his appearance. He wants his girlfriend to buy him a razor. None of the above.
What part of London is in France?	The letter N. The letter O. The letter L. None of the above.

Table 1: Example of sentence puzzle and word puzzle.

Adversarial Strategy	Question	Choice
Orignal	A man shaves everyday, yet keeps his beard long.	He is a barber. He wants to maintain his appearance. He wants his girlfriend to buy him a razor. None of the above.
Semantic Reconstruction	A man preserves a lengthy beard despite shaving every day.	He is a barber. He wants to maintain his appearance. He wants his girlfriend to buy him a razor. None of the above.
Context Reconstruction	Tom attends class every day but doesn't do any homework.	He is a teacher. He is a lazy person. His teacher will not let him fail. None of the above.

Table 2: Example of semantic reconstruction and context reconstruction.

2 Task Description

The BRAINTEASER QA task consists of two subtasks, sentence puzzles and word puzzles, as shown in Table 1. It requires awareness of common sense "default values" and covering them with unconventional thinking that distinguishes these default values from hard constraints.

Sentence Puzzle: Sentence-type brain teaser where the puzzle defying common sense is centered on sentence snippets.

Word Puzzle: Word-type brain teaser where the answer violates the default meaning of the word and focuses on the letter composition of the target question

It is worth noting that both tasks include an adversarial subset, created by manually modifying the original brain teasers without changing their latent reasoning path. In order to accurately evaluate the reasoning ability of our proposed system and ensure that it truly possesses lateral thinking ability, this task constructs adversarial versions of the original data in two ways:

Semantic Reconstruction: Rephrasing the original question without changing the correct answer and the distractors, as showing in table 2.

Context Reconstruction: Keeping the original reasoning path but changing both the question and the answer to describe a new situational context.

Finally, the task also proposes two evaluation metrics to ensure the accuracy of the system in both the overall test set and each adversarial subset. These two evaluation indicators are described as follows:

Instance-based Accuracy: Consider each issue (original/adversarial) in the test set as a separate instance to test the overall accuracy of the system's output on the test set.

Group-based Accuracy: Each question and its associated adversarial instances form a group, and a system will only receive a score of 1 when it correctly solves all questions in the group.

3 Methodology

We propose a system that simulates human lateral thinking patterns, which consists of two stages. During the first stage, our system engages in a simulation of how humans typically read and interpret Brainteaser question stems. The aim here is to check the question stems meticulously, intending to pinpoint specific elements that appear to contravene established common sense norms. The second stage is to combine the parts that violate common sense with four options for thinking, find the option that can "resolve" the parts that defy common sense, and use it as the final answer. The overall architecture of the system is shown in Figure 1.

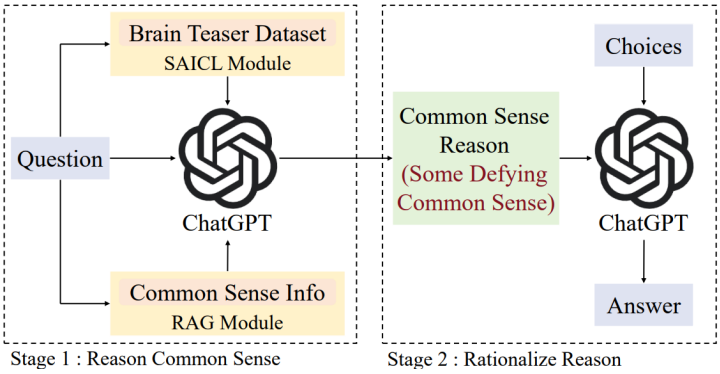


Figure 1: The overall architecture of our proposed system

3.1 Common Sense Reason

In this stage, we use a LLM as the core to conduct common sense reasoning. We input Brainteaser's problem directly into LLM and use LLM's powerful language ability to infer the unreasonable aspects of the problem. At the same time, in order to suppress the hallucination problem of LLM, we design two modules, namely the RAG module and the SAICL module.

3.1.1 Retrieval Augmented Generation

The RAG module integrates deep learning technologies such as Retrieval and Generation. This module is designed to enhance the LLM's understanding of input questions by retrieving relevant information from a vast array of unstructured documents as well as structured knowledge graphs, specifically for Brainteaser questions, in order to produce more accurate, richer, and more relevant Defying Common Sense Reasoning. The workflow of the RAG module includes the following two steps:

Retrieval Phase: Retrieve relevant information from a large number of unstructured documents and structured knowledge graphs according to the given Brainteaser problem.

Integration Phase: The retrieved information snippets are then integrated and merged to be effectively utilized by the generation model. This includes reordering, filtering, or encoding the retrieval results to better suit the subsequent generation tasks.

3.1.2 Self-Adaptive In-Context Learning

We are inspired by Wu et al. (Wu et al., 2023) and develop a SAICL module.The SAICL module adaptively selects better In Context example data from the training set for each Brainteaser problem to improve the effectiveness of In Context Learning. The workflow of the SAICL module also consists of two phases:

Selection Phase: Using the top-K method, search for the K question and its options and answers that are closest to the Brainteaser question in the semantic space.

Sorting Phase: Using the Minimum Description Length (MDL) principle to find an organization that minimizes the compressed encoding length of the output given the input and context. This can be represented by equation (1):

$$c^* = \arg\min_{c \in \mathbf{C}} L_\theta(y \mid c, \mathbf{x}) + L(\theta), \quad (1)$$

where each c represents one possible organization of examples. $L_\theta(y \mid c, \mathbf{x})$ is the code-length required to compress and transmit testing label y given the organization c and testing input $\mathbf{x}$. $L(\theta)$ is the code-length required to describe the model, and it can be calculated in the following equation (2):

$$L_\theta(y \mid c, \mathbf{x}) \approx -\mathbb{E}_{q(y_i|Y)} \log_2 p\left(y_i \mid c, \mathbf{x}\right), \quad (2)$$

where $q\left(y_i \mid Y\right)$ is the prior of y_i among all possible labels Y. Through the above calculation, further select a suitable subset from the K examples selected in the previous phase as the context examples for Brainteaser, combine them with the output of the RAG module, and input them into LLM.

By combining these two modules with the powerful language capabilities of LLM, we can derive reasonable yet contradictory common sense reasoning from the Brainteaser problem. For example, when our question is: "How could a cowboy ride into town on Friday, stay two days, and ride out on Wednesday?" We will gain some common sense reasoning as follows:

- Cowboy rides into town on Friday.
- Cowboy stays in town for two days.
- Cowboy rides out on Wednesday.
- Sunday is two days after Friday.

In this way, we can clearly see the defying common sense part in the Brainteaser question.

3.2 Rationalize Defying Common Sense

At this stage, we combine the conflicting reasoning obtained in the previous stage with the first three options of Brainteaser. This fusion is achieved through the careful design of specific prompt words, which are crafted with the express purpose of evaluating whether any of these three preliminary options possess the capability to logically reconcile the previously identified conflicting reasoning.

For example, in the example given in section 3.1, the option: "His horse is named Wednesday" can effectively solve the Defying Common Sense part inferred from the Common Sense Reason. So it is the correct answer.

But, it is crucial to highlight that in instances where none of the first three options succeeds in producing a satisfactory rationale that effectively addresses the contradictory reasoning, our model is programmed to adopt a fallback strategy. In such scenarios, the model is designed to automatically select the fourth option, aptly labeled "None of above". This decision-making protocol ensures that our model retains the flexibility to understand situations where the presented options fail to provide a coherent resolution to the discrepancies identified, thereby maintaining the integrity of our analytical process. This strategic approach underscores the meticulousness with which our system evaluates the available options, ensuring a comprehensive and reasoned determination of the most appropriate response.

4 Experimental Setup

In this section, we introduce our system settings, and baseline model.

4.1 System Settings

In the RAG module of our system SHTL, we initially remove the stop-words from the original Brainteaser question, then use ConceptNet to retrieve the meanings and relationships of the remaining parts, followed by deduplication and sorting based on relevance to the question. Subsequently, we design appropriate prompt words to concatenate them. In the SAICL module, during the search phase, we utilize the Bert model to obtain feature vectors for each question in the training set. In the vector space, we compare these vectors with the target question's feature vector using cosine similarity, selecting the ten most similar entries. During the ranking phase, we follow the method of the original paper (Wu et al., 2023),randomly select eight entries, extracting them sixteen times, and then calculate the score for each combination obtained from these extractions according to Section 3.1.2. We then select the best combination and use appropriate prompts to link them. At the end of the first stage, we use appropriate prompts to combine the results from both the RAG and SAICL modules with the original Brainteaser question and input them into ChatGPT, obtaining Defying Common Sense. This is then combined with the options of the Brainteaser question using appropriate prompts and input into ChatGPT to derive the best answer.

4.2 Baseline

Our baseline models are categorized into three types: one consists of Large Language Models with a minimal number of prompts, another incorporates models endowed with common sense knowledge, and finally, human evaluation.

Prompted Models:

We evaluate the instruction-finetuned LLMs in few-shot setting:

• **ChatGPT** It is one of the publicly available state-of-the-art Large Language Models in the GPT series (Brown et al., 2020).

• **T0** (Sanh et al., 2022) It is an LLM trained through multi-task instruction tuning, possessing strong zero-shot generalization capabilities.

• **FlanT5** (Chung et al., 2022) It is an enhanced version of T5 (Raffel et al., 2020).

To ensure a fair comparison with human performance, when prompting ChatGPT in a zero-shot setting, we add a description indicating that the question is a brain teaser requiring creative thinking for its resolution. For the other models, we employ the same instruction templates found in their training datasets.

Common Sense Models:

To understand the impact of common sense knowledge on our task, we evaluate the following models enhanced with common sense:

• **RoBERTa-L (CSKG)** (Ma et al., 2021) It is a model fine-tuned on synthetic QA pairs generated from various Common Sense Knowledge Graphs (CSKG) (Ilievski et al., 2021).

• **CAR**(Wang et al., 2023) It is a model finetuned in a similar pipeline as (Ma et al., 2021) but with enhanced negative sampling strategy and reportedly superior performance.

For reference, we also include the native RoBERTa model (Liu et al., 2019) to understand

Category	Model	Instance-based			Group-based		overall
		Original	Semantic	Context	Ori & Sem	Ori & Sem & Con	
Random		25.8	24.2	22.5	5.0	2.5	25.0
Sentence Puzzle							
Prompted Models	FlanT5(780M)	18.7	16.3	22.0	10.5	4.3	19.0
	FlanT5(3B)	26.8	25.4	35.4	20.1	12.9	29.2
	FlanT5(11B)	33.5	31.6	36.8	22.0	11.0	34.0
	T0(11B)	22.0	22.0	29.7	16.3	11.0	24.6
	T0P(11B)	23.9	22.5	34.9	17.7	12.0	27.1
	T0PP(11B)	26.3	27.3	37.8	19.1	12.0	30.5
	ChatGPT	60.8	59.3	67.9	50.7	39.7	62.7
Common Sense Models	RoBERTa-L	43.5	40.2	46.4	33.0	20.1	43.4
	RoBERTa-L(CSKG)	35.4	36.8	45.0	28.7	18.2	39.0
	CAR	10.5	10.5	11.5	5.7	2.4	10.9
Human		87.5	**90.0**	**95.0**	87.5	**87.5**	**90.8**
SHTL		**90.0**	**90.0**	87.5	**90.0**	**87.5**	89.2
Word Puzzle							
Prompted Models	FlanT5(780M)	22.6	17.7	28.7	9.1	3.7	23.0
	FlanT5(3B)	37.8	29.9	42.7	23.2	12.8	36.8
	FlanT5(11B)	42.7	32.9	43.9	28.7	20.1	39.8
	T0(11B)	17.1	14.0	23.2	9.8	6.1	18.1
	T0P(11B)	28.7	26.2	34.2	19.5	12.8	29.7
	T0PP(11B)	33.5	31.1	39.6	20.1	11.0	34.8
	ChatGPT	56.1	52.4	51.8	43.9	29.3	53.5
Common Sense Models	RoBERTa-L	19.5	19.5	23.2	14.6	6.1	20.7
	RoBERTa-L(CSKG)	18.9	16.5	30.5	12.8	6.1	22.0
	CAR	38.4	31.1	20.1	26.2	6.1	29.2
Human		84.4	87.5	**90.6**	84.4	**84.4**	**87.5**
SHTL		**90.6**	**93.8**	78.1	**90.6**	68.8	**87.5**

Table 3: Main zero-shot results over two BRAINTEASER subtasks across all models in all metrics, "Ori" is Original, "Sem" is Semantic and "Con" is Context. The best performance among all models is in bold.

the impact of common sense knowledge.

Human Evaluation:

We recruit four volunteers who are completely unfamiliar with our task to help us test the test set, and take the average of their results as the human test result.

5 Results and Analysis

The final results of our experiments are presented in Table 3. As can be seen from Table 3, the outcomes of the majority of Prompted Models as well as Common Sense Models are essentially random, and some are even below random performance. It is noteworthy to mention the ChatGPT model, which achieves a score of 62.7 in Sentence Puzzles and 53.5 in Word Puzzles, making it the best-performing model aside from Humans and our system, SHTL. From the evaluation results of Humans, it is evident that for both Sentence Puzzles and Word Puzzles, the Human Evaluation scores for Ori & Sem and Ori & Sem & Con were identical, indicating that human lateral thinking capabilities are remarkably stable and unaffected by the Adversarial Subset. Finally, our proposed system, SHTL, can surpass Human performance in most categories, with an average score in the two subtasks that is only 0.8 lower than that of Humans. This significantly exceeds the performance achieved using ChatGPT alone, suggesting that the latent linguistic capabilities of LLMs need to be further explorer appropriately.

6 Conclusion

In this paper, we introduce a lateral thinking system named SHTL, designed to simulate human lateral thinking capabilities for solving brain teaser questions. The system is divided into two stages. The first stage focuses on common sense reasoning, primarily comprised of the RAG module and the SAICL module, which are interconnected through appropriate prompt words to generate instances of defying common sense. The second stage involves identifying the correct options to rationalize the defying common sense generated in the previous stage. This system achieves competitive results, significantly outperforming the ChatGPT setting in a zero-shot scenario, and its performance on the test set is close to that of human evaluation.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Filip Ilievski, Pedro A. Szekely, and Bin Zhang. 2021. CSKG: the commonsense knowledge graph. In *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, volume 12731 of *Lecture Notes in Computer Science*, pages 680–696. Springer.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14317–14332. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13507–13515. AAAI Press.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Victor Sanh, Albert Webson, and Colin Raffel et al. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Weiqi Wang, Tianqing Fang, Wenxuan Ding, Baixuan Xu, Xin Liu, Yangqiu Song, and Antoine Bosselut. 2023. CAR: conceptualization-augmented reasoner for zero-shot commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 13520–13545. Association for Computational Linguistics.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1423–1436. Association for Computational Linguistics.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huan, and Tao Gui. 2023. The rise and potential of large language model based agents: A survey. *CoRR*, abs/2309.07864.

BadRock at SemEval-2024 Task 8: DistilBERT to Detect Multigenerator, Multidomain and Multilingual Black-Box Machine-Generated Text

Marco Siino
Department of Electrical, Electronic
and Computer Engineering
University of Catania
Italy
marco.siino@unipa.it

Abstract

The rise of Large Language Models (LLMs) has brought about a notable shift, rendering them increasingly ubiquitous and readily accessible. Across diverse platforms such as social media platforms, news outlets, educational platforms, question-answering forums, and even academic domains, there has been a notable surge in machine-generated content. Recent iterations of LLMs, exemplified by models like ChatGPT and GPT-4, exhibit a remarkable ability to produce coherent and contextually relevant responses across a broad spectrum of user inquiries. The fluidity and sophistication of these generated texts position LLMs as compelling candidates for substituting human labour in numerous applications. Nevertheless, this proliferation of machine-generated content has raised apprehensions regarding potential misuse, including the dissemination of misinformation and disruption of educational ecosystems. Given that humans marginally outperform random chance in discerning between machine-generated and human-authored text, there arises a pressing imperative to develop automated systems capable of accurately distinguishing machine-generated text. This pursuit is driven by the overarching objective of curbing the potential misuse of machine-generated content. Our manuscript delineates the approach we adopted for participation in this competition. Specifically, we detail the fine-tuning and the use of a DistilBERT model for classifying each sample in the test set provided. Our submission is able to reach an accuracy equal to 0.754 in place of the worst result obtained at the competition that is equal to 0.231.

1 Introduction

Large language models (LLMs) are increasingly pervasive and readily accessible, leading to a surge in machine-generated content across a multitude of platforms (Fang et al., 2024). LLMs have demonstrated an impressive ability to generate highly fluent responses to diverse user queries. The eloquent nature of these generated texts renders LLMs appealing candidates for replacing human labour across various scenarios. However, this widespread adoption has sparked concerns regarding the potential misuse of such texts, including the dissemination of misinformation in journalistic contexts and disruptions within educational systems (Tang et al., 2023).

The increasing adoption of Transformer-based architectures in academic research has also been bolstered by various methodologies showcased at SemEval 2024. These methodologies tackle diverse tasks and yield noteworthy findings. For instance, at the Task 2 (Jullien et al., 2024), where to address the challenge of identifying the inference relation between a plain language statement and Clinical Trial Reports is used T5 (Siino, 2024b); Task 4 (Dimitrov et al., 2024) and Task 10 (Kumar et al., 2024) where is employed a Mistral 7B model to detect persuasion techniques in memes (Siino, 2024a) and to perform Emotion Recognition in Conversation (ERC) within Hindi-English code-mixed conversations respectively (Siino, 2024c).

Despite human evaluators marginally outperforming random chance in distinguishing between machine-generated and human-written text (Mitchell et al., 2023), the need for automatic methods to detect machine-generated content has become increasingly urgent. This necessity prompted the organizers of Task 8 at SemEval-2024 to focus on developing such methods with the aim of mitigating potential misuse.

Previous efforts in detecting machine-generated text have been made. For instance, (Guo et al., 2023) devised methods to discern whether a text was generated by ChatGPT or authored by a human across various domains. However, these endeavours primarily concentrated on the outputs of ChatGPT.

The RuATD Shared Task 2022 tackled artificial

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 239–245

text in Russian, spanning models for paraphrase generation, text simplification, text summarization, and machine translation (Shamardina et al., 2022). However, their emphasis was on models fine-tuned for specific tasks or domains, which differs from the focus of the Task 8. While (Mitchell et al., 2023) detected outputs of various LLMs such as GPT-2, OPT-2.7, Neo-2.7, GPT-J, and NeoX, it's pertinent to note that these models have become obsolete with the advent of GPT-3 and even GPT-4. The Task 8 hosted at SemEval 2024 was built upon the previous work discussed in (Wang et al., 2023b).

To address these objectives, there is an ongoing demand for automated tools capable of extracting and categorizing data, facilitating the classification with recent NLP models. Recent advancements in the machine and deep learning architectures have spurred heightened interest in Natural Language Processing (NLP). Substantial endeavours have been directed towards devising techniques for the automated identification and categorization of textual content accessible on the internet today. In the literature, to perform text classification tasks, several strategies have already been proposed. In the last fifteen years, some of the most successful strategies have been based on SVM (Colas and Brazdil, 2006; Croce et al., 2022), on Convolutional Neural Network (CNN) (Kim, 2014; Siino et al., 2021), on Graph Neural Network (GNN) (Lomonaco et al., 2022), on ensemble models (Miri et al., 2022; Siino et al., 2022) and, recently, on Transformers (Vaswani et al., 2017; Siino et al., 2022b).

Participants in SemEval-2024 Task 8 could compete for three Subtasks better described in the rest of this paper. However, our team participated in the first Subtask only. The first Subtask (i.e., Subtask A) is the Binary Human-Written vs. Machine-Generated Text Classification one: Participants are tasked with determining, based on a given full text, whether it is human-written or machine-generated. There are two tracks for Subtask A: monolingual (only English sources) and multilingual.

The subsequent sections of the paper are structured as follows: Section 2 offers background information on Task 6, held at SemEval-2024. In Section 3, we outline the approach introduced in this study. Section 4 delves into the specifics of the experimental setup employed to reproduce our findings. The outcomes of the official task and relevant discussions are presented in Section 5. Finally, Section 6 concludes our study and suggests avenues for future research.

We make all the code publicly available and reusable on GitHub[1].

2 Background

The task 8 hosted at SemEval-2024 (Wang et al., 2024) is made of three different Subtasks.

- Subtask A. Binary Human-Written vs. Machine-Generated Text Classification: Given a full text, determine whether it is human-written or machine-generated. There are two tracks for Subtask A: monolingual (only English sources) and multilingual.
- Subtask B. Multi-Way Machine-Generated Text Classification: Given a full text, determine who generated it. It can be human-written or generated by a specific language model.
- Subtask C. Human-Machine Mixed Text Detection: Given a mixed text, where the first part is human-written and the second part is machine-generated, determine the boundary, where the change occurs.

An example from the original dataset - better described in (Wang et al., 2023b) - is the one that follows.

Given the prompt:

> *"Write a Wikipedia article with the title Ẅilliam Whitehouse̵, the article should at least have 250 words."*

While the original human-written version is available online[2], the output produced by a ChatGPT 3.5 Turbo was:

> *"William Whitehouse was a 19th-century British engineer and inventor who made significant contributions to the field of hydraulics. He was born in 1825 in Birmingham, England, where he grew up and eventually became an apprentice at the Soho Foundry. This apprenticeship instilled in him a passion for steam engines and hydraulics that would shape his future career. After completing*

[1] https://github.com/marco-siino/SemEval2024/

[2] https://en.wikipedia.org/wiki/William_Whitehouse

his apprenticeship, Whitehouse worked for the Birmingham Water Works Company as an engineer. It was during this time that he began experimenting with hydraulic machinery and invented several innovative devices. In 1850, he developed the first hydraulic press, which was able to exert immense amounts of pressure and became widely used in industrial manufacturing processes. He also created hydraulic lifts for raising and lowering heavy loads and hydraulic cranes for lifting and moving heavy objects. Whitehouse's inventions revolutionized manufacturing and construction industries and helped to increase productivity and efficiency. In addition to hydraulic machinery, he also worked on steam engines and locomotives, improving their performance and reliability. He was regarded as an expert in his field and wrote several papers on hydraulic engineering that were widely read and respected. Whitehouse's achievements did not go unnoticed by the scientific community, and he was awarded the Royal Society's prestigious Telford Medal in 1875. He also became a fellow of the Royal Society and was elected to the Institution of Civil Engineers and the Institution of Mechanical Engineers. Despite his success, Whitehouse remained humble and dedicated to creating innovative and practical solutions to engineering problems. He passed away in 1905, leaving behind a legacy of groundbreaking hydraulic inventions that continue to play a vital role in modern manufacturing and construction industries."

3 System Overview

Even if it has already been proved that the Transformers are not necessarily the best option for any text classification task (Siino et al., 2022a), depending on the goal, some strategies like domain-specific fine-tuning (Sun et al., 2019; Van Thin et al., 2023), or data augmentation (Lomonaco et al., 2023; Mangione et al., 2022; Siino et al., 2024a) can be beneficial for the considered task.

However, to address the Task 8 hosted at SemEval-2024 we employed a zero-shot learning strategy (Chen et al., 2023; Wahidur et al., 2024), making use of DistilBERT (Sanh et al., 2020), fine-tuned on the SST-2 dataset (Socher et al., 2013).

DistilBERT, akin to its larger counterparts (i.e., BERT), exhibits commendable performance across a diverse array of tasks when fine-tuned. While prior research predominantly delved into distillation techniques for crafting task-specific models, the distillation approach in this case harnesses knowledge distillation during the pre-training phase. DistilBERT demonstrate the feasibility of reducing the size of a BERT model by 40%, while retaining 97% of its language understanding prowess and achieving 60% increase in speed. To harness the inductive biases inherent in larger models during pre-training, a triple loss mechanism is introduced with this model. This mechanism combines language modelling, distillation, and cosine-distance losses. The compact, expedited, and resource-efficient model not only streamlines the pre-training process but also showcases its potential for on-device computations through a proof-of-concept experiment and comparative on-device analysis.

The Stanford Sentiment Treebank stands as the inaugural corpus equipped with fully labeled parse trees, facilitating comprehensive exploration of the compositional effects of sentiment in language. It comprises 11,855 individual sentences culled from film reviews. Leveraging the Stanford parser, the corpus encompasses a total of 215,154 unique phrases, each annotated by three human evaluators. This novel dataset affords an opportunity to delve into the intricacies of sentiment analysis and capture nuanced linguistic phenomena. Numerous examples within the corpus exhibit distinct compositional structures. The granularity and breadth of this dataset are poised to empower the community in training compositional models grounded in supervised and structured machine learning methodologies. While extant datasets primarily focus on document and chunk labelling, there remains a pressing need to enhance sentiment capture from concise remarks, such as those found in Twitter data.

Utilizing DistilBERT trained on the SST Stanford dataset for detecting human or AI-generated text holds significant promise due to its nuanced understanding of sentiment and context. By leveraging DistilBERT's fine-grained sentiment analysis capabilities, coupled with its proficiency in discerning contextual nuances, the model we used is supposed to effectively distinguish between human-

generated and AI-generated text. The SST dataset, annotated for human sentiments classification task, enables DistilBERT to grasp the subtleties of human language, making it adept at identifying deviations indicative of AI-generated content. Moreover, fine-tuning DistilBERT on this dataset enhances its sensitivity to linguistic cues that differentiate human-authored texts from those generated by AI algorithms, thereby offering a robust solution for text authenticity verification in various applications, including misinformation detection, content moderation, and forensic linguistics.

In this study, we employed a fine-tuning approach to enhance the performance of DistilBERT, initially trained on the SST dataset, for the task of distinguishing between human and AI-generated text. The fine-tuning process involved training the model for three epochs on the provided training set, utilizing a portion of the data for validation. Specifically, we partitioned 20% of the training set samples to form a validation set, crucial for assessing the model's performance and preventing overfitting. After completing the fine-tuning process, we systematically evaluated the model's performance across the three epochs on the validation set. Subsequently, we selected the tuned version of the model that exhibited superior performance, as determined by its validation set accuracy. This validation methodology ensures the reliability and generalization capability of the fine-tuned DistilBERT model for the targeted task of differentiating between human and AI-generated text.

In a recent study (Siino et al., 2024b), has been shown that the contribution of preprocessing for text classification tasks is generally not impactful when using Transformers. More specifically, the best combination of preprocessing strategies is not very different from doing no preprocessing at all in the case of Transformers. For these reasons, and to keep our system highly fast and computationally light, we have not performed any preprocessing on the text.

4 Experimental Setup

We implemented our model on Google Colab. The library we used comes from HuggingFace[3] and is the uncased version of DistilBERT specifically trained on the above-mentioned SST2 dataset [4]. We did perform a three-epochs additional fine-tuning, before generating the prediction on the unlabelled test set. This model is versatile and can serve as a foundational tool for topic classification tasks. While it can function as a raw model for masked language modelling or next sentence prediction, its primary utility lies in its adaptability for fine-tuning on downstream tasks. Users can explore the model hub to discover fine-tuned versions tailored for specific tasks beyond its original scope. As already mentioned, all of our code is available on GitHub.

5 Results

Given the binary nature of the classification task, the organizers proposed *Accuracy* as the evaluation metric to be considered for the final ranking. The accuracy is defined in the Equation 1. Where TP stands for the number of correctly predicted right answers, FP stands for the number of wrongly predicted right answers, and FN stands for the right answers wrongly predicted as wrong answers.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

In Table 1, we present the outcomes derived from our methodology. They are the same results publicly availble on the official final ranking shown on the official task page[5] and on CodaBench[6].

Compared to the best performing models, our simple approach exhibits some room for improvements. It is worth notice that required no further pre-training and the computational cost to address the fine-tuning stage is manageable with the free online resources offered by Google Colab. However, even with the low effort required, it is possible to achieve interesting results with our proposed approach. Out of the 137 participants, our approach, based on the use of a fine-tuned version of DistilBERT, is able to rank between the position 68 and 69 in the final ranking.

6 Conclusion

This paper presents the application of a DistilBERT-model for addressing the Task 8 at SemEval-2024.

[3] https://huggingface.co/

[4] https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english

[5] https://github.com/mbzuai-nlp/SemEval2024-task8

[6] https://www.codabench.org/competitions/1752/

TEAM NAME	Accuracy
safeai (1)	0.969
comp5 (2)	0.961
halwhat (3)	0.961
baseline (19-20)*	0.885
DistilBERT (68-69)*	0.754
saibewaraditya (137)	0.231

Table 1: Comparing performance on the test set. In the table are shown the results obtained by the first three teams, by the last one and by our approach. In parentheses is reported the position in the official final ranking. Our approach is not ranked in the official final ranking, but the score obtained ranks between the positions 68 and 69.

For our submission, we decided to fine-tune a pre-trained Transformer. The model was used to perform a sequence classification task to detect if a piece of text is written by a human or by a generative model. The task is challenging, and there is still opportunity for improvement, as can be noted looking at the final ranking. Possible alternative approaches to our can include utilizing the few-shot capabilities or also the use of other models like Llama and T5, eventually using further data, or directly integrating other samples from the training and from the development sets. Further improvements could be obtained with a fine-tuning and modelling the problem as a text classification task. Furthermore, given the interesting results recently provided on a plethora of tasks, also other few-shot learning (Wang et al., 2023a; Maia et al., 2024; Siino et al., 2023; Meng et al., 2024) or data augmentation strategies (Muftie and Haris, 2023; Tapia-Téllez and Escalante, 2020; Siino and Tinnirello, 2023) could be employed to improve the results. Looking at the final ranking, our simple approach exhibits some room for improvements. However, it is worth notice that it has required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab.

Acknowledgments

We extend our gratitude to the anonymous reviewers for their insightful comments and valuable suggestions, which have significantly enhanced the clarity and presentation of this paper.

References

Shiming Chen, Ziming Hong, Wenjin Hou, Guo-Sen Xie, Yibing Song, Jian Zhao, Xinge You, Shuicheng Yan, and Ling Shao. 2023. Transzero++: Cross attribute-guided transformer for zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12844 – 12861.

Fabrice Colas and Pavel Brazdil. 2006. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Daniele Croce, Domenico Garlisi, and Marco Siino. 2022. An SVM ensemble approach to detect irony and stereotype spreaders on twitter. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2426–2432. CEUR-WS.org.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Xiao Fang, Shangkun Che, Minjia Mao, Hongzhe Zhang, Ming Zhao, and Xiaohang Zhao. 2024. Bias of ai-generated content: an examination of news produced by large language models. *Scientific Reports*, 14(1):1–20.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29,*

2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1746–1751. ACL.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Francesco Lomonaco, Gregor Donabauer, and Marco Siino. 2022. COURAGE at checkthat!-2022: Harmful tweet detection using graph neural networks and ELECTRA. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 573–583. CEUR-WS.org.

Francesco Lomonaco, Marco Siino, and Maurizio Tesconi. 2023. Text enrichment with japanese language to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2708–2716. CEUR-WS.org.

Beatriz Matias Santana Maia, Maria Clara Falcão Ribeiro de Assis, Leandro Muniz de Lima, Matheus Becali Rocha, Humberto Giuri Calente, Maria Luiza Armini Correa, Danielle Resende Camisasca, and Renato Antonio Krohling. 2024. Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer. *Expert Systems with Applications*, 241:122418.

Stefano Mangione, Marco Siino, and Giovanni Garbo. 2022. Improving irony and stereotype spreaders detection using data augmentation and convolutional neural network. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2585–2593. CEUR-WS.org.

Zong Meng, Zhaohui Zhang, Yang Guan, Jimeng Li, Lixiao Cao, Meng Zhu, Jingjing Fan, and Fengjie Fan. 2024. A hierarchical transformer-based adaptive metric and joint-learning network for few-shot rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(3).

Mohsen Miri, Mohammad Bagher Dowlatshahi, Amin Hashemi, Marjan Kuchaki Rafsanjani, Brij B Gupta, and W Alhalabi. 2022. Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*, 37(12):11319–11341.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Fuad Muftie and Muhammad Haris. 2023. Indobert based data augmentation for indonesian text classification. In *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, page 128 – 132.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the the ruatd shared task 2022 on artificial text detection in russian. *arXiv preprint arXiv:2206.01583*.

Marco Siino. 2024a. Mcrock at semeval-2024 task 4: Mistral 7b for multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024b. T5-medical at semeval-2024 task 2: Using t5 medical embeddings for natural language inference on clinical trial data. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024c. Transmistral at semeval-2024 task 10: Using mistral 7b for emotion discovery and reasoning its flip in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2021. Detection of hate speech spreaders using convolutional neural networks. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 2126–2136. CEUR-WS.org.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2022a. Fake news spreaders detection: Sometimes attention is not all you need. *Information*, 13(9):426.

Marco Siino, Marco La Cascia, and Ilenia Tinnirello. 2022b. Mcrock at semeval-2022 task 4: Patronizing and condescending language detection using multi-channel cnn, hybrid lstm, distilbert and xlnet. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 409–417. Association for Computational Linguistics.

Marco Siino, Francesco Lomonaco, and Paolo Rosso. 2024a. Backtranslate what you are saying and i will tell who you are. *Expert Systems*, n/a(n/a):e13568.

Marco Siino, Maurizio Tesconi, and Ilenia Tinnirello. 2023. Profiling cryptocurrency influencers with few-shot learning using data augmentation and ELECTRA. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2772–2781. CEUR-WS.org.

Marco Siino and Ilenia Tinnirello. 2023. Xlnet with data augmentation to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2763–2771. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2022. T100: A modern classic ensemble to profile irony and stereotype spreaders. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2666–2674. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024b. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*.

José Medardo Tapia-Téllez and Hugo Jair Escalante. 2020. Data augmentation with transformers for text classification. In *Advances in Computational Intelligence*, pages 247–259, Cham. Springer International Publishing.

Dang Van Thin, Duong Ngoc Hao, and Ngan Luu-Thuy Nguyen. 2023. Vietnamese sentiment analysis: An overview and comparative study of fine-tuning pretrained language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Rahman S. M. Wahidur, Ishmam Tashdeed, Manjit Kaur, and Heung-No Lee. 2024. Enhancing zero-shot crypto sentiment with fine-tuned language model and prompt engineering. *IEEE Access*, 12:10146 – 10159.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023a. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Trans. Circuits Syst. Video Technol.*, 33(12):7789–7802.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

WarwickNLP at SemEval-2024 Task 1: Low-Rank Cross-Encoders for Efficient Semantic Textual Relatedness

Fahad Ebrahim
The University of Warwick
Coventry, United Kingdom
Fahad.Ebrahim@warwick.ac.uk

Mike Joy
The University of Warwick
Coventry, United Kingdom
M.S.Joy@warwick.ac.uk

Abstract

This work participates in SemEval 2024 Task 1 on Semantic Textural Relatedness (STR) in Track A (supervised regression) in two languages, English and Moroccan Arabic. The task consists of providing a score of how two sentences relate to each other. The system developed in this work leveraged a cross-encoder with a merged fine-tuned Low-Rank Adapter (LoRA). The system was ranked eighth in English with a Spearman coefficient of 0.842, while Moroccan Arabic was ranked seventh with a score of 0.816. Moreover, various experiments were conducted to see the impact of different models and adapters on the performance and accuracy of the system.

1 Introduction

Semantic Textual Relatedness (STR) is a measure of how closely related or connected two linguistic units are in terms of their meanings or concepts (Abdalla et al., 2023). STR is a valuable concept in Natural Language Processing (NLP), as it helps us to understand the connections and similarities between different pieces of text. By determining the degree of relatedness between sentences or phrases, we can improve various NLP tasks such as information retrieval, question answering, and text summarisation. This understanding of semantic relatedness enables us to create more accurate word embeddings and sentence representations, enhancing the performance of language processing models. One way to represent STR is a supervised regression task in which the output is a continuous score number between 0 and 1.

For the STR task in SemEval 2024 (Ousidhoum et al., 2024b), the organisers on Track A (supervised) provided datasets (Ousidhoum et al., 2024a) for nine languages or dialects. They provided pairs of sentences and annotated the degree of relatedness via a human score between 0 and 1. The languages considered in this work are English and Moroccan Arabic. These two were selected because they are comprehensive for the team.

There are various methods that can be used to estimate the relatedness of two sentences. One of the methods is to utilise the Pre-Trained Language Models (PLMs). PLMs are currently state-of-the-art in the field of NLP, and follow the transformer architecture introduced in (Vaswani et al., 2017) with the attention mechanism. Another variation that uses a mechanism of cross-attention is the cross encoder (Reimers and Gurevych, 2019), whichthat of takes two inputs and outputs a score between 0 and 1 on how related these two inputs are. They are efficient in determining the correlation between two inputs.

Parameter-efficient fine-tuning (PEFT) aims to tune the pre-trained model with high accuracy but with less cost and complexity. One of the PEFT methods is adapters (Poth et al., 2023), which tune extra parameters or layers instead of tuning the whole model while maintaining competitive accuracy. They can be considered as few-shot learners as per (Beck et al., 2022). One type of adapter is the low-rank adapter (LoRA). Instead of tuning the whole weight, LoRA adds small matrices in each layer, and these matrices would be fine-tuned.

Hence, this work applies a tuned LoRA adapter on a pre-trained cross-encoder to estimate the score of the relatedness of two sentences. The code is publicly available[1].

The paper is organised as follows: Section 2 presents the background, including related work and dataset overview; Section 3 covers the system overview; Section 4 presents the results; Section 4 discusses the error analysis and limitations; and the paper concludes.

[1] https://github.com/FahadEbrahim/STR_LoRA

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 246–252

2 Background

This section will cover the related work, dataset description and a brief introduction to PEFT.

2.1 Related Work

There have been previously related versions of the STR datasets, such as (Asaadi et al., 2019) and (Abdalla et al., 2023). Different versions use different languages, annotations or available datasets.

Another related dataset is Semantic Textual Similarity (STS) (Cer et al., 2017). There are differences between STS and STR. Specifically, STS tasks aim to assess how similar two text segments or sentences are, focussing on tasks such as identifying paraphrases or entailment relationships. On the other hand, STR looks at the overall closeness in meaning between linguistic units, considering various factors such as topic-relatedness and stylistic similarities. Thus, STR is more general, and STS can be treated as a subset of STR. Secondly, the outputs differ slightly, as the output of the previous STS tasks is between zero and five, while in STR, the output is between zero and 1. STS datasets were beneficial in this task, as can be seen later in the paper.

The task of STS is a common natural language understanding task. Several approaches have been used for STS. One of the popular approaches utilises the generation of robust contextual embeddings and then uses a similarity measurement like cosine similarity to get the required score. The embeddings can be extracted with a Universal Sentence Encoder (USE) (Cer et al., 2018), Language-agnostic BERT Sentence Embedding (LabSE) (Feng et al., 2022) or Sentence BERT (SBERT) (Reimers and Gurevych, 2019). These are different approaches to get meaningful embeddings that can capture the semantics of the input sentences.

2.2 Dataset

The datasets (Ousidhoum et al., 2024a) provided for training consist of two sentences and a score of how related they are between 0 and 1. Sample instances of the English training dataset can be seen in Table 1. The first example shows two sentences with the same meaning, and therefore the score is 1. The second example shows partially related sentences with a score of 0.5. The last example includes two unrelated sentences with a score of around 0.

Sentences	Score
Actor Gazzara dead at 81 Actor Ben Gazzara dies at 81	1.0
yeah and so is bubbles lol Bubbles used to reside next door	0.5
A child wielding a snow shovel. A cat bites a human's nose.	0.03

Table 1: Dataset training sample instances.

The datasets are split into 3 sets: training, development, and testing. The number of instances in each set in the English and Moroccan Arabic languages can be seen in Table 2. The main reason for injecting adapters directly is that there are few training samples in Moroccan Arabic. So, few-shot learning is a better approach for this language and, therefore, for the overall task.

Set/Language	English	Moroccan Arabic
Train	5500	925
Evaluation	250	70
Testing	2500	427

Table 2: STR Dataset split.

2.3 Parameter Efficient Fine-Tuning

PLMs follow the Encoder/Decoder architecture introduced by Transformer (Vaswani et al., 2017). Models such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019), T5 (Raffel et al., 2020), and DeBERTa (He et al., 2020) are other variations of the transformer architecture.

Another variation that uses cross-attention is the cross-encoder (Reimers and Gurevych, 2019), which takes two inputs and outputs a score between zero and one. A cross-encoder trained on the STS-B benchmark (Cer et al., 2017) would result in an output score between 0 and 1 instead of 0 to 5.

One of the PEFT techniques is the use of adapters, which are efficient few-shot learners as per (Poth et al., 2023). There are various adapter architectures. Instead of tuning the entire model weights, the adapters would tune additional parameters, layers or weight matrices. The three types of adapters investigated in this work are Houlsby (Houlsby et al., 2019), Pfeiffer (Pfeiffer et al., 2021), and LoRA (Hu et al., 2021). The Houlsby adapter adds two additional layers before and after the feed-forward (FF) layer in each encoder, while

Pfeiffer adds only a single layer after the FF layer. The LoRA adapter adds small weight matrices in each layer of the transformer layers.

This work tunes several adapters on different pre-trained models and checks which combination performs best. The best architecture will be explained in the next section.

3 System Development

This section covers the cross-encoder, LoRA adapter, the developed system architecture, and the evaluation metric.

3.1 Cross-Encoder

The cross-encoder takes two inputs simultaneously and uses the concept of cross-attention allowing the model to capture interactions between the two sentences. The cross-encoder would generate a score between 0 and 1 indicating how similar the two inputs are. The technical architecture of the cross-encoder can be seen in Figure 1. The cross-encoder adds two special tokens: SEP to separate the two sentences and CLS. The CLS token gets added at the beginning of the concatenated sentences along with the SEP token and represents a classification vector. The initial CLS token identifies the representation of two sentences. The final CLS token captures the cross-attention between all previous CLS tokens and produces one final semantic vector. A classification head maps the final CLS vector to a score between 0 and 1 based on the chosen model.

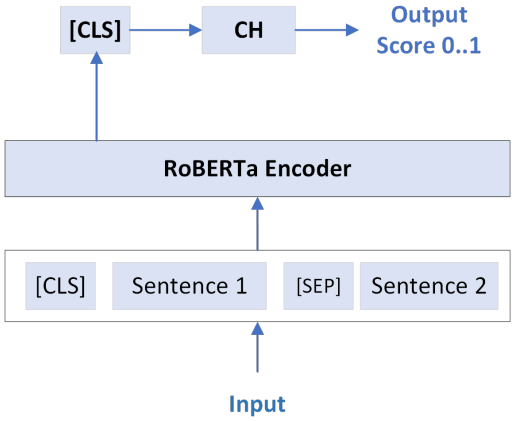


Figure 1: Cross-Encoder Architecture.

3.2 LoRA Adapter

The LoRA adapter (Hu et al., 2021) adds two additional low-rank matrices that are trainable instead of training the whole model. To explain LoRA mathematically, assume the input to a neural network to be X and the output to a single hidden layer is $h(x)$, then the output with full fine-tuning would equal the input multiplied by a weight matrix W_0 as per Equation 1. The weight matrix W_0 belongs to the dimension of $(d * k)$.

$$h(x) = W_0 X \qquad W_0 \in \mathbb{R}^{d \times k} \tag{1}$$

In LoRA, an additional weight matrix ΔW_0 is added into the input and initial weight matrices to get the hidden layer output as per equation 2. It belongs to the same dimension as the original matrix.

$$h(x) = W_0 X + \Delta W_0 X \qquad W_0, \Delta W_0 \in \mathbb{R}^{d \times k} \tag{2}$$

The new matrix is decomposed into two trainable matrices, B and A, with dimensions $(d * r)$ and $(r * k)$, respectively. The value of r (rank) is much smaller than d and k ($r \ll d, r \ll k$), so the new two matrices are smaller than the initial matrix. This reduces the model's trainable parameters while maintaining high accuracy. The value of r is a hyper-parameter and it is used as 8 in this work.

$$h(x) = W_0 X + (BA)X \qquad B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k} \tag{3}$$

Applying LoRA can reduce the size of the model from hundreds of megabytes to just a few megabytes (Hu et al., 2021) while maintaining a high level of accuracy.

3.3 System Architecture

The architecture of the system developed for the STR task can be seen in Figure 2. The following are the simplified steps.

1. Initialise the adapter with random weights.
2. The LoRA adapter is trained given the training sets. During training, only the low-rank matrices are fine-tuned.
3. The adapter merges with the classification head of the cross-encoder.
4. The testing data are fed into the cross-encoder with the attached adapter to get the relatedness score.

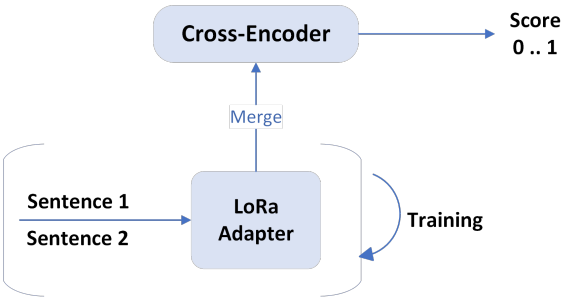


Figure 2: System Overview.

3.4 Evaluation

The evaluation is based on Pearson's correlation coefficient as in Equation 4. The values of X_i and Y_i represent individual instances while the values $\bar{X}$ and $\bar{Y}$ represent the mean of X and Y. The higher the coefficient, the better the model evaluation. The value of the coefficient ranges between -1 to 1 where 1 represents a higher correlation between the inputs. The metric value is generated with submissions being uploaded through CodaLab (Pavao et al., 2023). CodaLab is a platform for competitions and research purposes. The platform returns the scores for development and testing.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \quad (4)$$

In addition to the datasets, the organisers provided a baseline for all languages based on LabSE (Feng et al., 2022), which provides sentence embeddings. Scores were calculated based on cosine similarity between these embeddings.

4 Results

The system would initially be trained on the training set and evaluated with the development set, and the official results would be produced by applying the model to the testing sets. This work's approach is incremental and experimental. So, variations of models and adapters would be tested, and the models achieving better metrics would be selected.

4.1 Experimental Development Results

All the initial experiments were conducted in English. Then, the best approach was applied to Moroccan Arabic. The experiments carried out in the development phase were incremental in 10 epochs. Firstly, several models (BERT, ALBERT, RoBERTa, DeBERTa, and T5) were fine-tuned with the Pfeiffer adapter and the model with the best-yielding scores was selected. The models and their respectable Spearman scores can be seen in Table 3. The model with the highest score was RoBERTa, with a development score of 0.8155. The exact model names in Huggingface (Wolf et al., 2019) are available in Appendix A. Huggingface is a platform with a large number of pre-trained models. The initial selected models were general and not domain-specific. Therefore, a model trained for similarity could be investigated. The chosen model was a cross-encoder trained with the STS-B dataset. The results were better than the base RoBERTa model, reaching a development score of 0.8296. So, the system continued using this model.

Model	Score
BERT	0.8023
ALBERT	0.7755
DeBERTa small	0.8088
T5 small	0.8003
RoBERTa base	0.8155
RoBERTa STSB-CE	0.8296

Table 3: Selected models and their development scores.

Then, the impact of merging several single adapters (Houlsby, Pfeiffer and LoRA) on the cross-encoder was studied. Table 4 shows the scores with the attachment of the three adapters. The LoRA adapter scored the highest with a Spearman coefficient of 0.8343. To further improve the accuracy, the training epochs were increased to 30 to improve the generalisation of the model, and this was achieved by producing a final development score of 0.8417.

Adapter	Score
Pfeiffer	0.8296
Houlsby	0.8309
LoRA	0.8343

Table 4: Impact of different adapters on the RoBERTa STSB-CE in the development phase.

The same configuration was used for the Moroccan Arabic language, resulting in a development score of 0.8577.

To see whether an Arabic model would perform differently in an Arabic pre-trained model, another experiment in Moroccan Arabic was conducted in the development set using the CAMeLBERT model (Inoue et al., 2021). This model was trained

on a large corpus of Arabic for different tasks with several dialects (modern standard Arabic, dialectal Arabic, and classical Arabic). There is also a version of the CAMeLBERT model trained on the combination of the three dialects. This version produced a higher score of 0.871 on the Moroccan Arabic development set. However, this result was not submitted for the official competition.

4.2 Official Results

The same configuration used in the development phases was applied in the test set. The model used for both languages was the cross-encoder RoBERTa trained on STS-B. The adapter was LoRA and the training epochs were 30. The rest of the parameters can be seen in Appendix B. The official results reported can be seen in Table 5. The scores for English and Moroccan Arabic were 0.8425 and 0.8163 respectively. Both exceed the baseline scores of 0.83 and 0.77 respectively.

System	Baseline	Our System
English	0.83	0.8425
Morrocan Arabic	0.77	0.8163

Table 5: Official submitted scores of the competition in the English and Morrocan Arabic.

5 Discussion

5.1 Error Analysis

One interesting negative result was found during development. The developed system was applied to Algerian Arabic, which yielded a low evaluation metric of 0.54. This could be attributed to the fact that this dialect had majorly unseen data for the model. The STS-B dataset has some overlap with the two languages (English and Moroccan Arabic) worked on in this paper.

The development set with labels was not utilised in the testing phase. Moreover, the differences between applying the adapters on the cross-encoder are minor. So, applying these models to the testing set may yield different results.

The results on CAMelBERT on Moroccan Arabic were better than the cross-encoder in the development phase, but this was not considered in the system's official results to maintain the consistency of the used model (cross-encoder). The usage of CAMelBERT yielded a metric value of 0.8347, which is higher than the cross-encoder. This was noticed in the post-evaluation phase and, therefore, not reported in the official results. This indicates that using a pre-trained model on a large corpus of Arabic yields a better result than the cross-encoder trained on the STSB dataset.

5.2 Limitations

Due to time constraints, this work has several limitations. Firstly, the models were not fully fine-tuned with the training data. It would be comprehensive to compare the full fine-tuning to the adapter tuning and discuss the obtained results. Secondly, there was no hyper-parameter optimisation was done in this system except for the number of epochs. The effect of applying different parameters to different models would yield a better overview of the effect of these hyper-parameters on the training process. Thirdly, full training of the cross-encoder would be an interesting scope of future work. Lastly, there was no pre-processing, and the sentences were tokenised as they were. The impact of pre-processing could be studied.

5.3 Ethical Statement

This work used only the STS-B and STR datasets, which are both publicly available. Although the dataset has some overlap between the development and testing sets, this work maintains that the development set was not used for training. This is attributed to the ethical convention in machine learning of using the appropriate set for the suitable task.

Conclusion

This work involves participating in the SemEval 2024 STR task in two languages (English and Moroccan Arabic). The regression task is to predict a score relating to two sentences. The developed system consists of a LoRA adapter trained on the given training instances and then attached to a cross-encoder previously trained on the STS-B dataset. The system achieved excellent performance, exceeding the baseline and ranking seventh in Moroccan Arabic and eighth in English, with Pearson Coefficient scores of 0.8163 and 0.8425, respectively.

Acknowledgements

The authors thank the organisers for answering questions about the task and express their gratitude to the anonymous reviewers for their constructive feedback.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796.

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. 2019. Big bird: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 505–516.

Tilman Beck, Bela Bohlender, Christina Viehmann, Vincent Hane, Yanik Adamson, Jaber Khuri, Jonas Brossmann, Jonas Pfeiffer, and Iryna Gurevych. 2022. AdapterHub playground: Simple and flexible few-shot learning with adapters. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–75, Dublin, Ireland. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association*

for Computational Linguistics: Main Volume, pages 487–503, Online. Association for Computational Linguistics.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. 2023. Adapters: A unified library for parameter-efficient and modular transfer learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160, Singapore. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

A Model Names

The exact model names in Huggingface[2] used in this work are available in Table 6.

Table 6: Model Names in Huggingface.

Model	Model Name
BERT	google-bert/bert-base-uncased
RoBERTa	FacebookAI/roberta-base
DeBERTa	microsoft/deberta-v3-small
ALBERT	albert/albert-base-v2
T5	google-t5/t5-small
RoBERTaCE	cross-encoder/stsb-roberta-base
CAMeLBERT	CAMeL-Lab/ bert-base-arabic-camelbert-mix

[2] https://huggingface.co/

B Hyper-parameters Configuration

The hyper-parameters used to develop the system are available in Table 7. The number of epochs during the development phase was 10, while in testing, it was increased to 30.

Parameter	Value
Epoch	10 - 30
Learning rate	5×10^{-4}
Batch size	30
LoRA (r and alpha)	8
GPU	Tesla A100 (40GB)

Table 7: Parameter configuration.

C Used Libraries

This section includes the Python libraries used in the code and their versions according to Table 8.

Library	Version
adapters	0.1.1
transformers	4.35.2
datasets	2.17.0
sklearn	1.2.2
torch	2.1.0
accelerate	0.27.0
pandas	1.5.3
numpy	1.23.5

Table 8: Used Python libraries.

NU-RU at SemEval-2024 Task 6: Hallucination and Related Observable Overgeneration Mistake Detection Using Hypothesis-Target Similarity and SelfCheckGPT

Thanet Markchom[1] and **Subin Jung**[2] and **Huizhi Liang**[2]
[1]Department of Computer Science, University of Reading, Reading, UK
[2]School of Computing, Newcastle University, Newcastle upon Tyne, UK
t.markchom@pgr.reading.ac.uk, {s.jung4, huizhi.liang}@newcastle.ac.uk

Abstract

One of the key challenges in Natural Language Generation (NLG) is "hallucination", in which the generated output appears fluent and grammatically sound but may contain incorrect information. To address this challenge, "SemEval-2024 Task 6 - SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes" is introduced. This task focuses on detecting overgeneration hallucinations in texts generated from Large Language Models for various NLG tasks. To tackle this task, this paper proposes two methods: (1) hypothesis-target similarity, which measures text similarity between a generated text (hypothesis) and an intended reference text (target), and (2) a SelfCheckGPT-based method to assess hallucinations via predefined prompts designed for different NLG tasks. Experiments were conducted on the dataset provided in this task. The results show that both proposed methods can effectively detect hallucinations in LLM-generated texts.

1 Introduction

Natural Language Generation (NLG) is a field within Natural Language Processing (NLP) that focuses on enabling machines to produce human-like texts. In NLG, one of the challenges is the phenomenon of "hallucination", where the generated output is fluent and grammatically sound but contains incorrect information or extends beyond the provided information. This issue is particularly significant in NLG applications where correctness is crucial, such as machine translation and paraphrasing. It can compromise the quality and reliability of the generated content, resulting in a loss of fidelity to the sources or models from which the content is generated. To address this challenge, "SemEval-2024 Task 6 - SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes" (Mickus et al., 2024) is introduced. This task aims to identify grammatically correct outputs that contain incorrect semantic information or overgenerated content, with or without access to the model that produced the output. The outputs are obtained from various Large Language Models (LLMs) in three distinct NLG tasks: definition modeling (DM), machine translation (MT), and paraphrase generation (PG).

Recent efforts have been made to develop frameworks for detecting hallucinations in LLM-generated texts. One approach involves calculating information overlap and contradictions between generated and reference texts (Dhingra et al., 2019; Shuster et al., 2021). Higher mismatches suggest a greater likelihood of hallucination. Another popular approach is an LLM-based evaluation. This approach focuses on prompting LLMs to assess a machine-generated text and determine the probability of this text being a hallucination (Kadavath et al., 2022; Manakul et al., 2023).

Despite the success of these existing methods, they have mainly focused on detecting factual hallucinations. This paper further explores how information overlap calculation and LLM-based evaluation approaches can be applied to detect overgeneration hallucinations. Specifically, we propose two methods to detect overgeneration hallucinations in SemEval Task 6. The first method is hypothesis-target similarity, which measures text similarity between a generated text (hypothesis) and an intended reference text (target). The second method is an LLM-based evaluation approach that utilizes a state-of-the-art framework called SelfCheckGPT (Manakul et al., 2023). This method assesses hallucinations via distinct predefined prompts tailored for texts generated from different NLG tasks.

2 Related Work

Recently there have been some attempts to develop frameworks for evaluating hallucinations in LLM-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 253–260

generated texts. One approach is to consider lexical features of LLM-generated texts and reference texts and calculate the information overlap and contradictions between the generated and the reference texts. The higher the mismatch counts, the lower the faithfulness and thus the higher the hallucination score. For example, Dhingra et al. (2019) proposed PARENT (Precision And Recall of Entailed n-grams from the Table), which is capable of assessing hallucinations by referencing both the source and target texts. Shuster et al. (2021) introduced a metric for knowledge-grounded dialogue tasks aimed at measuring the alignment between LLM-generated texts and the relevant knowledge judged by humans. Martindale et al. (2019) proposed the Bag-Of-Vectors Sentence Similarity (BVSS) metric for assessing sentence adequacy in machine translation. This metric aids in identifying disparities in information between the output and the translation reference. Despite the simplicity and effectiveness of information overlapping, it has limitations in handling syntactic or semantic variations, which can impact its accuracy in evaluating faithfulness.

Another recent approach is an LLM-based evaluation where an LLM is prompted to evaluate generated texts, e.g., to predict the probability that a generated text is a hallucination. For instance, Kadavath et al. (2022) used LLMs to evaluate the validity of their own claims by asking models to first generate answers and then to evaluate the probability that their answers are correct. Manakul et al. (2023) proposed an approach called SelfCheckGPT with prompts. In their approach, each LLM-generated sentence was compared against multiple generated responses from an LLM. An LLM was asked to assess whether an LLM-generated sentence was supported by the generated responses. If it was consistently supported by multiple responses, then it was likely to not be a hallucination. Friel and Sanyal (2023) proposed the ChainPoll approach where an LLM was asked to decide whether an LLM-generated text contained hallucinations, using a detailed and carefully engineered prompt. However, the majority of existing approaches have primarily focused on detecting factual hallucinations related to incorrect information in texts, rather than overgeneration hallucinations. Thus, there remains a critical need to explore and adapt these approaches for the detection of overgeneration hallucinations.

3 Problem Formulation

The objective of this task is to predict whether the actual model production (generated text) is a hallucination, with or without having access to the model that generated the text. Specifically, each input in this task consists of the following information:

- Task (task): the task for which the model was optimized, which can be either Definition Modeling (DM), Paraphrase Generation (PG), or Machine Translation (MT).
- Source (src): the input provided to the model.
- Target (tgt): the intended reference 'gold' text that the model is expected to generate.
- Hypothesis (hyp): the actual model output.
- Reference (ref): specifies whether the target, source, or both fields contain the semantic information necessary to establish whether the hypothesis is a hallucination.
- Model Checkpoint (model): Identifies the model used to produce the hypothesis (only applicable for model-aware inputs).

For each input, the goal is to predict a label indicating whether the hypothesis is a hallucination, along with the probability of the hypothesis being a hallucination (p(Hallucination)).

In this task, two datasets were provided: **model-aware dataset** and **model-agnostic dataset**. In the model-aware dataset, model checkpoints (available on HuggingFace) were provided for every sample. Conversely, in the model-agnostic dataset, these checkpoints were not included. For each dataset, an unlabeled training set, a validation set (with true labels), and a test set were provided. Also, a trial set was given without categorizing the samples based on whether they were model-aware or model-agnostic. The validation, trial and test sets contain binary annotations provided by at least five different annotators, along with a majority vote gold label.

4 Methods

To achieve the task of detecting overgeneration hallucinations, we propose two methods: (1) hypothesis-target similarity and (2) SelfCheckGPT-based methods. The details of each approach are discussed in the following subsections.

4.1 Hypothesis-Target Similarity Method

The proposed hypothesis-target similarity approach is an intuitive method for evaluating whether a generated text (hypothesis) contains hallucinations by comparing it with an intended reference or gold text (target). Specifically, we compute the text similarity between a hypothesis and a target and use the resulting value to determine whether the hypothesis contains hallucinations. The lower the similarity, the more likely it is that the hypothesis may contain a certain degree of hallucination. To compute text similarity, a text embedding method is first applied to generate embeddings of the generated and intended reference texts. In this work, we adopt *SentenceTransformers*[1] (Reimers and Gurevych, 2019) (*paraphrase-MiniLM-L6-v2*) to generate such embeddings since it has demonstrated success across various applications (Reimers and Gurevych, 2020; Choi et al., 2021; Markchom et al., 2020). Then, a cosine similarity metric is applied to these embeddings to compute the similarity. It is worth noting that other metrics are also applicable. This work selects cosine similarity due to its widespread usage and simplicity (Lin et al., 2014; Zhang et al., 2023).

After obtaining the similarity between a hypothesis and a target, we set a threshold δ to determine whether a hypothesis is a hallucination or not. If the similarity is lower than δ, it means that the hypothesis is different from the target and may contain hallucinations. Mathematically, given a hypothesis h and a target t, let e_h and e_t denote embeddings of the hypothesis and target, respectively. We define a function $f(h, t)$ that outputs the labels "Hallucination" and "Not Hallucination" for a given hypothesis h and target t as follows:

$$f(h,t) = \begin{cases} \text{Hallucination}, & \text{if } s(e_h, e_t) < \delta \\ \text{Not Hallucination}, & \text{otherwise} \end{cases} \quad (1)$$

where $s(e_h, e_t)$ denotes the cosine similarity between the hypothesis h and the target t. Furthermore, we compute $p(\text{Hallucination})$ based on the computed cosine similarity by applying a sigmoid function to the similarity as follows:

$$p(\text{Hallucination}) = \sigma(s(e_h, e_t)) \quad (2)$$

where σ denotes a sigmoid function. This function scales the computed similarity to the $[0, 1]$ interval and treats the resulting value as the probability of the hypothesis being a hallucination. Note that, in the PG task, target texts are unavailable for certain samples. Consequently, we consider source texts as target texts in these instances. In other words, we assess the similarity between a generated (paraphrased) text and its corresponding source text instead.

4.2 SelfCheckGPT-Based Method

In the SelfCheckGPT-based method, we adopt the SelfCheckGPT with Prompt approach in (Manakul et al., 2023) and design prompts to validate hallucinations. Specifically, for each sample, a prompt is crafted to assess whether a hypothesis is supported by a context, which includes a provided source and target (if available). If a hypothesis is not supported by a context, it is considered a hallucination. The prompt formats vary slightly for each task. Table 1 shows the prompt formats for samples from each task, where {src} denotes a source, {tgt} denotes a target, {hypo} denotes a hypothesis, and {term} denotes the term to be defined in a source only for the DM task. As shown in this table, the prompt for the DM task is noticeably different from the others. This is because we would like to semantically use the term as additional information apart from the source and hypothesis.

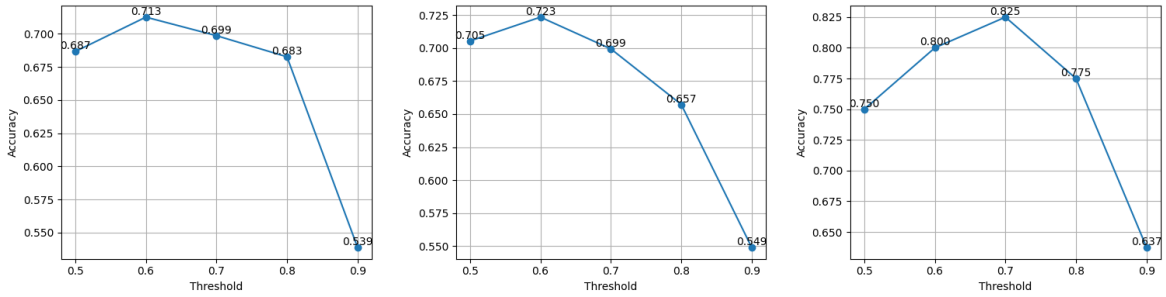
Task	Prompt format
DM	Context: {src} The term "{term}" means {tgt} Sentence: The term {term} means {hypo} Is the sentence supported by the context above? Answer Yes or No:
PG	Context: {src} Sentence: {hypo} Is the sentence supported by the context above? Answer Yes or No:
MT	Context: {src} {tgt} Sentence: {hypo} Is the sentence supported by the context above? Answer Yes or No:

Table 1: Prompt formats for samples from each task where {src} represent a source, {tgt} represents a target, {hypo} represents a hypothesis, and {term} represents the term to be defined in a source of the DM task.

Each prompt is run through the GPT-3.5 model (*gpt-3.5-turbo-1106*)[2] (Brown et al., 2020; OpenAI, 2022) N times, and the final label is determined by the majority of these responses. The probability $p(\text{Hallucination})$ of each sample is computed based

[1] https://www.sbert.net/

[2] https://platform.openai.com/docs/models/gpt-3-5



(a) Model-aware validation set (b) Model-agnostic validation set (c) Trial set

Figure 1: The accuracy on (a) the model-aware validation set, (b) model-agnostic validation set, and (c) the trial set when different threshold values are applied in the hypothesis-target similarity method

on the corresponding N responses as follows:

$$p(\text{Hallucination}) = \frac{1}{N}\sum_{i=1}^{N} l_i \qquad (3)$$

where l_i denotes the ith predicted label (1 for "Hallucination" and 0 for "Not Hallucination") identified from the ith response.

5 Experiments

The experiments were conducted on both **model-aware** and **model-agnostic** datasets. For each of these datasets, we applied the proposed methods to both the validation and test sets to evaluate their performance. Two evaluation metrics were employed: the *accuracy* of binary classification and the *Spearman correlation* of the predicted probabilities (p(Hallucination)) with the proportion of the annotators marking the hypothesis as "Hallucination". We compared the proposed methods with the baseline provided by the task organizers on the test set. This baseline is based on the SelfCheckGPT with Prompt approach, employing an open-source Mistral model (Jiang et al., 2023). In this baseline, for samples from a PG task, only the source text is provided as context to the Mistral model, similar to our SelfCheckGPT-based method. For DM and MT tasks, the baseline utilizes only the target text as context. In contrast, our SelfCheckGPT method incorporates both source and target texts as context. Also, in this baseline, each prompt was run through the Mistral model only once.

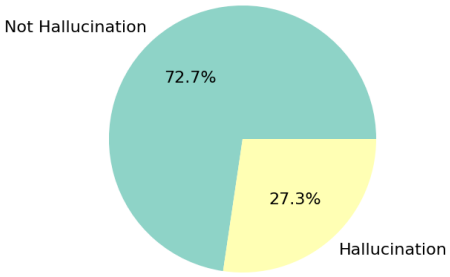
5.1 Hyperparameter Settings

Hypothesis-Target Similarity Method To determine the threshold δ, we conducted an analysis on the validation set to identify the optimal value. We varied the threshold from 0.5 to 0.9, increasing it by 0.1 at each step, and evaluated the accuracy on the validation and trial sets. Figure 1 displays the accuracy on the model-aware validation set, model-agnostic validation set, and the trial set when different threshold values were applied. From this figure, a threshold of 0.6 achieved the highest accuracy on the validation sets and closely approached the highest accuracy on the trial set. Therefore, we selected $\delta = 0.6$ when applying this method to the test set. To further examine the performance of using $\delta = 0.6$, it was applied to determine hallucinations on both the model-aware and model-agnostic training sets. However, since the training set is unlabelled, it is not possible to examine the accuracy. Therefore, our focus shifted to examining the frequency of "Hallucination" and "Not Hallucination" predictions. This was to ensure that using $\delta = 0.6$ would not result in the tendency of exclusively predicting one or the other. As shown in Figure 2, with $\delta = 0.6$, 27.3% and 41.3% of the samples in the model-aware and model-agnostic sets, respectively, were predicted as hallucinations.

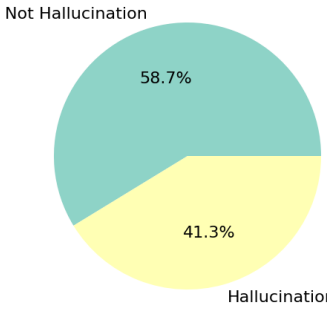
SelfCheckGPT-Based Method To select the number of generated responses (N), we varied N from 1 to 5. The accuracy and Spearman correlation results on both model-aware and model-agnostic validation sets, with different values of N, are presented in Figure 3. This figure indicates that as N increased, accuracy generally improved with fluctuations observed in both datasets. However, Spearman correlation consistently increased with no fluctuations as N increased. Therefore, we set N to 5 to obtain five responses for each sample in the test set. All hyperparameters of *gpt-3.5-turbo-1106* were configured with their default values. For any model response that indicated undetermined answers, the corresponding sample was considered

Dataset	Method	Validation set		Test set	
		Accuracy	Spearman correlation	Accuracy	Spearman correlation
Model-aware	Baseline	0.707	0.461	0.745	0.488
	Hypothesis-Target Similarity	0.699	**0.536**	0.734	0.518
	SelfCheckGPT-based	**0.722**	0.510	**0.768**	**0.582**
Model-agnostic	Baseline	0.649	0.380	0.697	0.403
	Hypothesis-Target Similarity	0.699	**0.574**	0.687	0.467
	SelfCheckGPT-based	**0.707**	0.567	**0.728**	**0.595**

Table 2: Comparative performance of the proposed methods measured by accuracy and Spearman correlation on the validation and test sets, with the highest value in bold

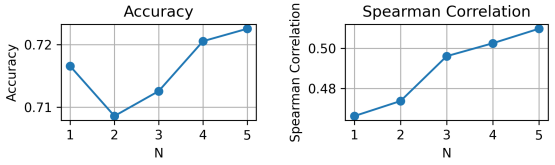


(a) Model-aware unlabelled training set with 0.6 threshold

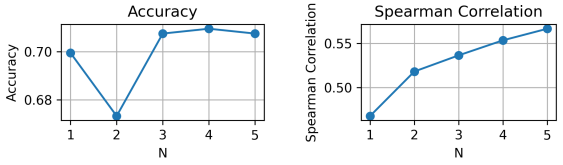


(b) Model-agnostic unlabelled training set with 0.6 threshold

Figure 2: The percentage of "Hallucination" and "Not Hallucination" from the result of the hypothesis-target similarity method

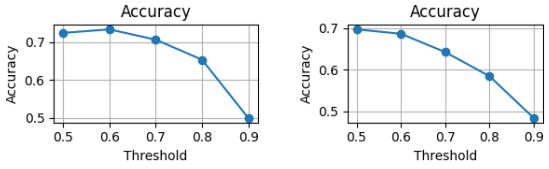


(a) Model-aware validation set



(b) Model-agnostic validation set

Figure 3: Accuracy and Spearman correlation results on both (a) model-aware and (b) model-agnostic validation sets when using different values of N.



(a) Model-aware (b) Model-agnostic

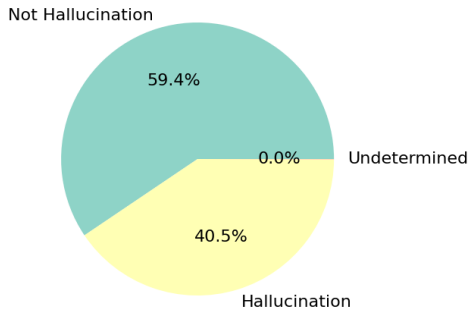
Figure 4: Accuracy results on (a) model-aware and (b) model-agnostic test sets using different thresholds δ.

as "Hallucination".

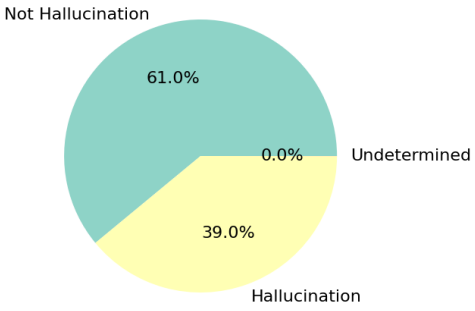
5.2 Results and Discussions

Table 2 shows the comparative performance of the proposed methods measured by accuracy and Spearman correlation on the validation and test sets. From this table, the proposed method based on SelfCheckGPT outperformed the baseline in terms of both accuracy and Spearman correlation on both model-aware and model-agnositc datasets. This indicates the effectiveness of using the GPT-3.5 model with prompts that include both source and target as context. Also, it suggests the benefit of running each prompt through an LLM multiple times to obtain a final prediction.

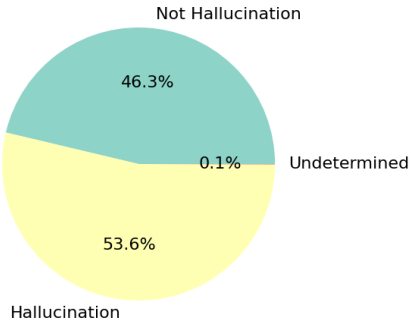
The proposed hypothesis-target similarity method closely approached the performance of the SelfCheckGPT-based approach on the validation sets, showing higher Spearman correlation values. However, on the test sets, the latter surpassed it. The reason could be that the selected threshold might not be precisely suitable for the test sets. Figure 4 shows the accuracy results on model-aware and model-agnostic test sets when different thresholds δ were used. From this figure,



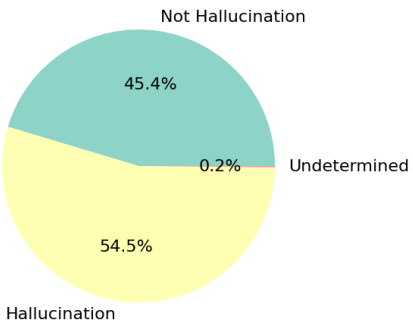
(a) Model-aware validation set



(b) Model-aware test set



(c) Model-agnostic validation set



(d) Model-agnostic test set

Figure 5: The percentages of response types, including "Not Hallucination", "Hallucination", and "Undetermined", obtained from the proposed SelfCheckGPT-based method on (a) model-aware validation set, (b) model-aware test set, (c) model-agnostic validation set, and (d) model-agnostic test set

using $\delta = 0.6$ resulted in the highest accuracy on the model-aware test set. However, on the model-agnostic test set, using $\delta = 0.5$ resulted in better accuracy. This indicates the challenge of selecting an optimal threshold based solely on observed data for generalizing to unseen data.

We investigated the number of undetermined responses in the SelfCheckGPT approach to validate whether this approach can effectively generate definitive answers for this task. Figure 5 shows the percentages of response types, including "Not Hallucination", "Hallucination", and "Undetermined", obtained from the proposed SelfCheckGPT-based method on the model-aware validation set, model-aware test set, model-agnostic validation set, and model-agnostic test set. According to this figure, the proposed SelfCheckGPT approach predicted less than 0.2% of undetermined answers across all datasets. This indicates that the SelfCheckGPT approach is effective in terms of producing definitive answers. Nonetheless, one limitation of this approach is its reliance on the availability of prior knowledge or expected outcomes (which, in this case, are the targets). In real-world situations, such information may not be available.

In the official competition rankings, the top-performing model achieved an accuracy of 0.813 and a Spearman correlation of 0.699 on the model-aware test set, and an accuracy of 0.847 and a Spearman correlation of 0.770 on the model-agnostic test set. Consequently, our SelfCheckGPT model secured the 26th position on the model-aware test set and the 35th position on the model-agnostic tes set.

6 Conclusions

This work proposes two methods for detecting hallucinations and observable overgeneration mistakes in texts generated by LLMs. The first method, the hypothesis-target similarity method, involves calculating the information overlap between a generated text and a reference text. This method utilizes a pre-trained SentenceTransformer model to calculate text embeddings for both the generated and reference texts, and cosine similarity to measure their similarity. The second method employs an LLM-based evaluation approach. It uses the SelfCheckGPT technique with prompts tailored to LLM-generated texts from various NLG tasks. The experimental results highlight the effectiveness of the proposed hypothesis-target similarity method in detecting hallucinations, particularly

when the similarity threshold is carefully chosen. Additionally, the findings reveal that the proposed SelfCheckGPT-based method outperformed the baseline, and effectively identified hallucinations in texts generated by LLMs. Moreover, these results underscore the significance of prompt design in evaluating hallucinations using LLMs. However, there is still room for improvement in the performance of our methods.

For future work, other SentenceTransformers models, such as Multi-QA or MSMARCO Passage models (SBERT.net, 2022) or alternative embedding models, such as InferSent (Conneau et al., 2018) or Universal Sentence Encoder (Cer et al., 2018) for the Hypothesis-Target Similarity approach will be considered. As for the SelfCheckGPT-based approach, other LLMs besides GPT-3.5 will also be investigated. Moreover, various prompt formats and the use of few-shot examples in the prompt will be explored.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder.

Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2021. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5482–5487.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2018. Supervised learning of universal sentence representations from natural language inference data.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv:1906.01081*.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *arXiv preprint arXiv:2310.18344*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Yung-Shen Lin, Jung-Yi Jiang, and Shie-Jue Lee. 2014. A similarity measure for text classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):1575–1590.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Thanet Markchom, Bhuvana Dhruva, Chandresh Pravin, and Huizhi Liang. 2020. UoR at SemEval-2020 task 4: Pre-trained sentence transformer models for commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 430–436, Barcelona (online). International Committee for Computational Linguistics.

Marianna Martindale, Marine Carpuat, Kevin Duh, and Paul McNamee. 2019. Identifying fluently inadequate output in neural and statistical machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 233–243, Dublin, Ireland. European Association for Machine Translation.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI. 2022. Gpt-3.5 turbo. `https://platform.openai.com/docs/models/gpt#gpt35-turbo`.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation.

SBERT.net. 2022. Pretrained models. https://www.sbert.net/docs/pretrained_models.html.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Ruichen Zhang, Zeshui Xu, and Xunjie Gou. 2023. Electre ii method based on the cosine similarity to evaluate the performance of financial logistics enterprises under double hierarchy hesitant fuzzy linguistic environment. *Fuzzy Optimization and Decision Making*, 22(1):23–49.

NCL_NLP at SemEval-2024 Task 7: CoT-NumHG: A CoT-Based SFT Training Strategy with Large Language Models for Number-Focused Headline Generation

Junzhe Zhao[1] and **Yingxi Wang**[2] and **Huizhi Liang**[3] and **Nicolay Rusnachenko**[3]
[1]Hangzhou Zero Matrix Intelligence Co., Ltd., China
[2]Huawei Technologies Co., Ltd., China
[3]School of Computing, Newcastle University, Newcastle upon Tyne, UK
zhaojunzhe_bit@163.com wangyingxiclaire@163.com
huizhi.liang@newcastle.ac.uk, rusnicolay@gmail.com

Abstract

Headline Generation is an essential task in Natural Language Processing (NLP), where models often exhibit limited ability to accurately interpret numerals, leading to inaccuracies in generated headlines. This paper introduces CoT-NumHG, a training strategy leveraging the Chain of Thought (CoT) paradigm for Supervised Fine-Tuning (SFT) of large language models. This approach is aimed at enhancing numeral perception, interpretability, accuracy, and the generation of structured outputs. Presented in SemEval-2024 Task 7 (task 3): Numeral-Aware Headline Generation (English), this challenge is divided into two specific subtasks. The first subtask focuses on numerical reasoning, requiring models to precisely calculate and fill in the missing numbers in news headlines, while the second subtask targets the generation of complete headlines. Utilizing the same training strategy across both subtasks, this study primarily explores the first subtask as a demonstration of our training strategy. Through this competition, our CoT-NumHG-Mistral-7B model attained an accuracy rate of 94%, underscoring the effectiveness of our proposed strategy, detailed in our project repository[1].

1 Introduction

Headline Generation is a key task in the field of Natural Language Processing (NLP), aimed at condensing the content of a given article into a concise, accurate, and information-rich single-sentence headline. This process requires not only an understanding of the article's core content but also the ability to creatively express this content (Matsumaru et al., 2020). Recently, Huang et al. (2023) conducted an in-depth analysis of the application of models (Lewis et al., 2019; Liu et al., 2022; Raffel et al., 2020; Wang et al., 2022a; Zhang et al., 2020) in the task of headline generation, revealing limitations of these models in processing numerical information. They identified that inaccuracies in the use of numbers significantly contribute to errors in generated headlines, a particularly critical issue in news headline generation where numbers often carry key information. To further explore the issue of numerical accuracy in news headlines, Huang et al. (2023) introduced a new dataset, NumHG, focused on the accuracy of numerical usage within news headlines. Their analysis revealed that news headline generation typically involves nine distinct methods for handling numbers—Copy, Translate, Round, Paraphrase, Add, Subtract, Divide, Multiply, and Span—each varying in complexity. These techniques enhance the interpretability and clarity of the headline generation process, showcasing a sophisticated blend of precision and creativity in distilling numerical information. Based on these insights, Chen et al. (2024) designed two independent tasks: the first requires models to mask numbers in given news articles and their headlines, then to accurately predict the masked numbers; the second involves generating news headlines with accurate numerical information based on the provided news content.

In the domain of NLP, Large Language Models (LLMs) have gained recognition for their capability to execute a wide array of tasks, including text generation, summarization, and question answering, using straightforward instructions. This demonstrates their remarkable versatility (Guo et al., 2023; Sahoo et al., 2024). To further enhance the adaptability of LLMs, fine-tuning techniques (Zhang et al., 2023) have been extensively applied, improving model performance on specific tasks while preserving a wide scope of application. LLMs typically utilize a decoder-only architecture (Radford et al., 2018) and adopt primarily two strategies for task-specific challenges: prompt engineering and fine-tuning. Prompt engineering enables the direct execution of tasks

[1]https://github.com/GavinZhao19/CoT-NumHG

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 261–269

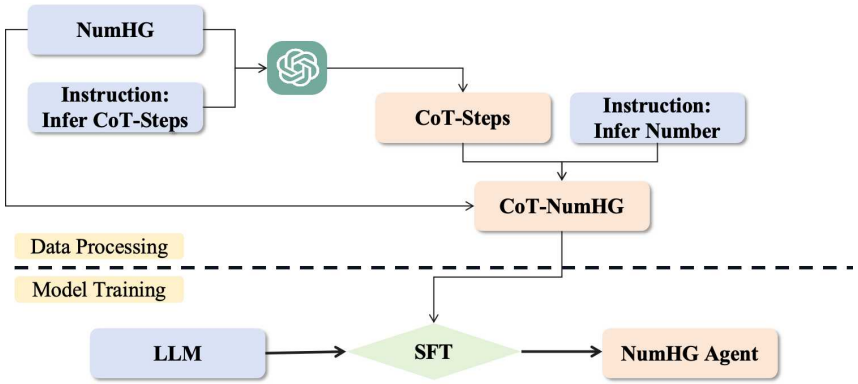


Figure 1: CoT-Based SFT Training Strategy Framework: The framework comprises two main parts: data processing and model training. In the first phase, data processing involves two steps. The first step combines specific instructions with the NumHG dataset, and through knowledge distillation using GPT-3.5-Turbo, new CoT-Steps are generated. These steps are then integrated with the corresponding instructions and the original dataset to produce the CoT-NumHG dataset. In the second phase, the CoT-NumHG dataset is utilized for the full-parameter SFT of the base model.

through various techniques, such as zero-shot (Radford et al., 2019), few-shot (Brown et al., 2020), chain of thought (CoT) (Wei et al., 2022), CoT with self-consistency (Wang et al., 2022b), and tree of thought (Yao et al., 2024), without requiring additional training. This underscores the models' ability to quickly adapt to new tasks by leveraging existing knowledge. Fine-tuning, via further training, refines the models' performance on specific tasks, particularly through Supervised Fine-Tuning (SFT) methods. To improve SFT efficiency, Parameter-Efficient Fine-Tuning (PEFT) techniques (Hu et al., 2023) including LoRA (Hu et al., 2021), prompt-tuning (Lester et al., 2021), and prefix-tuning (Li and Liang, 2021) have been introduced. These significantly enhance the models' adaptability and the quality of outputs for specific tasks without substantially increasing the model size or computational demands. This approach not only preserves the versatility of LLMs but also boosts their output quality and the ability to generate structured outputs in specific domains. Despite these approaches achieving certain levels of performance enhancement, there remains room for improvement in perceiving numerical information, reasoning ability, and generating structured outputs (Ouyang et al., 2023). Particularly in the task of news headline generation, reliance solely on prompt engineering may lead to uncontrollable outputs and insufficient structuring. Meanwhile, SFT, despite its ability to improve performance, shows limitations in the interpretability of the reasoning process and suffers from attention decay, potentially leading to the omission of important information.

To address these challenges, we propose a training strategy based on the CoT approach, designed to significantly enhance LLMs in the task of number-focused headline generation. Our method consists of two key components. First, drawing on the concept of knowledge distillation (Dasgupta et al., 2023), we utilize GPT-3.5-Turbo (Brown et al., 2020) and instructions to process the original NumHG dataset, generating a series of reasoning steps. Given the issue of attention decay when handling long-distance information (Xiao et al., 2023), we created a new CoT-NumHG dataset by combining the question statement with reasoning steps. This process aims to bolster the model's attention mechanism and improve the interpretability of the reasoning process (Wang et al., 2023). Secondly, we selected three LLMs as base models and performed full- parameters SFT using the constructed CoT-NumHG dataset on these base models. Through this approach, we not only significantly improved performance on the specific task, but also optimized structured outputs while maintaining the models' versatility. Our research contributions are threefold:

1. Based on the NumHG dataset, we developed the CoT-NumHG dataset, enhancing model interpretability and structured output capabilities. Importantly, we introduce a dataset construction technique specifically designed for the CoT-NumHG.

2. We demonstrate the enhancement of model performance through task-oriented SFT train-

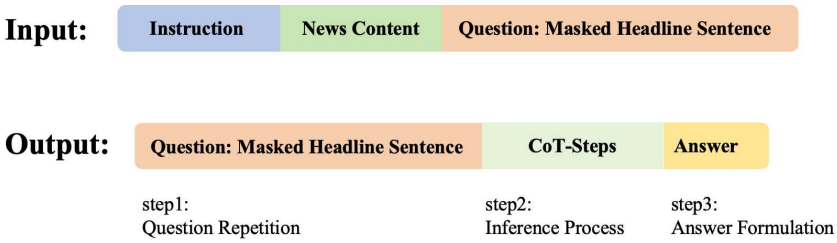


Figure 2: CoT-NumHG Dataset: The input of the dataset consists of three parts: Instruction, News Content, and Masked Headline Sentence (Question). The output is comprised of three components: Question Repetition, Inference Process, and Answer Formulation.

ing on the CoT-NumHG dataset across three base models, significantly improving news headline generation while maintaining general-purpose capabilities.

3. Through ablation studies, we demonstrated that the CoT-based training strategy effectively boosts the model's performance.

2 CoT-Based SFT Training Strategy Design

The training strategy of this study is divided into two main parts, as shown in Figure 1: the construction of the CoT-NumHG dataset and model training. Initially, through a knowledge distillation strategy, we enhanced the original dataset to improve the model's interpretability in handling the task of generating news headlines. Subsequently, the selected base LLMs were trained using full-parameter SFT techniques to achieve performance optimization and structured output for specific tasks.

2.1 CoT-Dataset Generation

To enhance the model's understanding of the relationships among news content, headline sentences, and answers, we employed a knowledge distillation approach during the data construction and optimization phases. Utilizing the original NumHG dataset and instructions, we generated inference processes through the GPT-3.5-Turbo model, termed CoT-Steps. CoT-Steps consist of three steps:

Step 1: **Identifying the Relevant Information**: This involves analyzing semantic relevance to pinpoint sentences in news articles that are closely related to the masked headline sentences and answers. This step ensures that the selected sentences are crucial for understanding the content of the news articles and for generating headlines.

Step 2: **Interpreting the Numerical Information**: For each identified key sentence, its direct numerical relevance to the generation task and the reasons for its selection are interpreted.

Step 3: **Choosing and Applying the Math Method**: For the numerical information in key sentences, appropriate methods are used for transformation and completion to accurately reflect in the generated headline sentences while maintaining logical consistency and accuracy.

This approach aims to bolster the model's data understanding and information processing capabilities by emulating the human problem-solving thought process, thereby enhancing attention scores and interpretability. Then, We integrated the reasoning process (CoT-Steps) into the training set to build a dataset specifically for SFT. Figure 2 shows that the input of this training set includes the instruction, news content, and the masked headline sentence; the output covers question restatement, CoT-Steps, conversion methods, and the final answer. This design aims to train the model to generate answers following given logical steps, thereby improving the accuracy and reliability of the generated results.

2.2 Model Training

For the model training part, we selected three large language models with a decoder-only architecture: ChatGLM3-6B (Du et al., 2022), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Zephyr-7B-Beta (Tunstall et al., 2023), for full-parameter fine-tuning. These models were chosen for their outstanding performance in text generation and comprehension, as illustrated in benchmarks comparing their capabilities to other LLMs of similar size (Zheng et al., 2023). During the fine-tuning

process, we focused on enhancing the models' comprehensive understanding and generation capabilities, especially in handling news headlines that contain numerical information. Through meticulous training methods, we ensured that the models could achieve higher performance on specific tasks.

3 Data Construction

3.1 CoT-Steps Generation

The primary source of the NumHG dataset is Newser[2], a news aggregation platform that provides headline news from both American and international media. News articles typically contain between 200 and 300 words. The entire NumHG dataset consists of news articles with titles that integrate numerical information, comprising 21,157 news articles for training and 2,572 for validation, totaling 23,729 articles. The data includes four keys: news, masked headline sentence, answer, and calculation. Initially, we employ a few-shot approach to distill the reasoning steps. The complete prompt given to the model comprises three parts: instruction, news, masked sentence (question), calculation, and answer. Figure 3 shows the instruction content used.

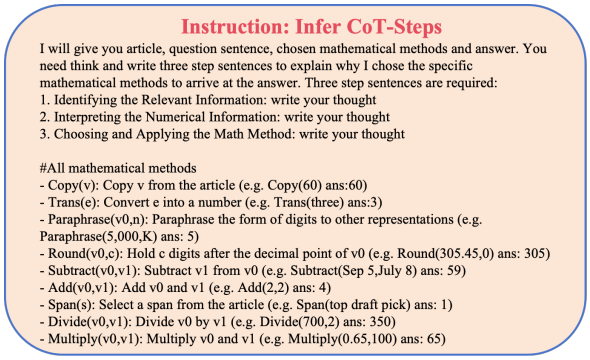


Figure 3: Instruction Prompt of Inferring CoT-Steps

Table 1 lists the detailed examples from the NumHG dataset, along with the inference steps obtained using knowledge distillation techniques. These steps not only reveal the key reasoning pathways in the news headline generation process but also provide clear guidance for models to more effectively handle numerical information and generate structured headlines.

3.2 CoT-NumHG Datasets Generation

In preparing the CoT-NumHG datasets for model training, we have adopted an approach inspired by the methodologies outlined in the "Never Lost in the Middle" study (Junqing et al., 2023). This strategy ensures that the model can efficiently identify and utilize key information within extended texts. Our dataset is specifically tailored to enhance the models' attention mechanisms, thereby improving their reasoning capabilities and their ability to produce structured outputs for complex tasks.

Table 1: Example of NumHG datasets and CoT-Steps

News: (Apr 18, 2016 1:02 PM CDT) Ingrid Lyne, the Seattle mom allegedly murdered while on a date, left behind three daughters—and a GoFundMe campaign set up to help the girls has raised more than $222,000 so far, Us reports. A friend of the family set up the campaign, and says that all the money raised will go into a trust for the girls, who are ages 12, 10, and 7. Lyne's date was charged with her murder last week.
Masked Headline (Question): $____K Raised for Kids of Mom Dismembered on Date
Calculation: Paraphrase(222,000,K)
Answer: 222
CoT Steps: 1. Identifying the Relevant Information: The relevant information in this question is the amount of money raised for the kids of the mom who was dismembered on a date. 2. Interpreting the Numerical Information: The numerical information given is $222,000. 3. Choosing and Applying the Math Method: I chose the Paraphrase method to convert the numerical information from the form of digits to other representations. By paraphrasing 222,000 as K, I am representing the amount as 222 thousand dollars.

The dataset (see the example in Table 4) is meticulously organized, comprising three elements in its input section: instruction (as referenced in Figure 4), news content, and masked headline sentences (posed as questions). This configuration is designed to keep the model focused during the processing of information and to encourage a logical and structured approach to output generation. In the output section, we employ a stepwise methodology to formulate answers. Initially, the model is instructed to repeat the question, a step that not only deepens its understanding of the query, but also counteracts attention drift by enhancing attention scores. Following this, the model boosts the interpretability of its reasoning process by executing CoT-Steps, which involve generating a sequence of intermediate reasoning steps. These steps are designed to mimic human problem-solving processes, thereby clarifying the model's reasoning pathway. Ultimately, the model presents the final answer, ensuring the creation of a structured and precise headline while preserving the integrity of the news content. Through this dataset design, our

[2] https://www.newser.com/

objective is for the model to demonstrate enhanced accuracy and interpretability in news headline generation tasks, in addition to maintaining consistent performance when managing information over long distances.

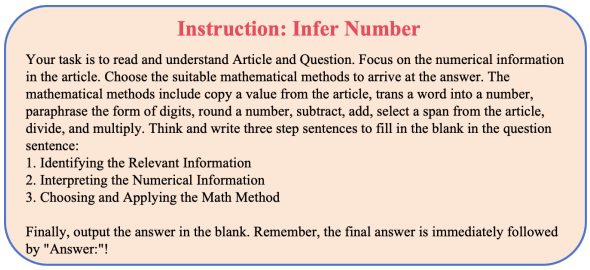


Figure 4: Instruction Prompt of Inferring Number

By incorporating these strategies into our dataset design, we aim to equip models with the ability to achieve superior accuracy and interpretability, particularly in tasks such as news headline generation, while also ensuring steady performance in the processing of long-range information. This methodology embodies the proverb "the pen is mightier than memory," emphasizing the importance of a structured and considered compilation of training data to bolster model capabilities in the NLP field.

4 Experiments and Results

4.1 Experiments Settings

Ultimately, we constructed the CoT-NumHG dataset, which comprises 23,729 samples. To ensure the randomness and fairness of the experiments, the dataset was randomly shuffled and 3,000 samples were selected as the validation set, leaving 20,729 samples for the training set. We chose three state-of-the-art LLMs as our base models: ChatGLM3-6B, Mistral-7B-Instruct-v0.2, and Zephyr-7B-Beta. SFT was performed on these base models.

For each base model, training was conducted over 2 epochs on 16 $\times$ A100 GPUs (80GB). During training, we set the batch size to 32, the learning rate to $1 \cdot 10^{-6}$, and the warmup ratio to 0.01. A cosine annealing schedule was used for learning rate adjustment, and training was accelerated using `bfloat16` precision. Cross-entropy loss was used to compute the loss. We monitored the evaluation loss (eval loss) of each model on the validation set during training. ChatGLM3-6B achieved the lowest eval loss (0.176) at epoch 1.95, with a corresponding training loss (train loss) of 0.121. Mistral-7B-Instruct-v0.2 reached its lowest eval loss (0.153) at epoch 1.81, with a training loss of 0.111. Zephyr-7B-Beta achieved the lowest eval loss (0.151) at epoch 1.96, with a training loss of 0.110.

Table 2: Example of CoT-NumHG-Mistral-7B's output

News: (Jul 20, 2009 3:41 PM CDT) The special inspector overseeing Treasury's TARP program says federal assistance to banks and other financial entities could end up costing taxpayers $23.7 trillion, Bloomberg reports. Aside from the $700 billion bailout, Neil Barofsky says in testimony prepared for told Congress tomorrow, other trillion-dollar federal programs could balloon...
Masked Headline (Question): TARP Watchdog: Bailouts May Cost US $____T
CoT-NumHG-Mistral-7B Output: To fill in the blank in the question sentence: TARP Watchdog: Bailouts May Cost US $____T. 1. Identifying the Relevant Information: The relevant information is the cost of the bailouts, which is given as $23.7 trillion. 2. Interpreting the Numerical Information: The numerical information is in the form of a decimal, with two digits after the decimal point. 3. Choosing and Applying the Math Method: I chose the math method Round(23.7,0) to round the number to the nearest whole number.This is because the question asks for the answer to be filled in the blank as a whole number, and rounding 23.7 to the nearest whole number gives us 24. Summary: math methods: Round(23.7,0) Answer: 24

To ensure the accuracy of the results, we performed ablation studies. Specifically, we used only the instruction prompt (also CoT) to generate outcomes with three base models, along with a benchmark model, GPT-3.5-Turbo. Additionally, we trained these three base models solely with the NumHG dataset, comparing the results against those trained using the CoT-NumHG dataset. The vLLM framework (Kwon et al., 2023) was consistently employed for inference.

4.2 Results

The competition provided a dataset containing 4,921 samples. The results showed that the first-place participant achieved an accuracy of 95%, while the participants in second to fourth places all reached an accuracy of 94%. Our team's submission, the CoT-NumHG-Mistral-7B model, also achieved an accuracy of 94% in this task, demonstrating the effectiveness of the CoT training strategy in enhancing model performance. Table 2 presents an example of the result.

To further analyze model performance, we conducted ablation studies across all models. We observed incremental improvements in accuracy, start-

ing from models prompted solely by CoT, progressing through those trained on the NumHG dataset, and culminating with those trained on the CoT-NumHG dataset. The accuracy of the CoT-NumHG-Mistral-7B model increased from 0.58 to 0.94, surpassing the untrained baseline, and improved from 0.73 to 0.94 compared to NumHG-Mistral-7B, showcasing significant improvements. This indicates that the CoT-Based SFT training strategy not only enhances model accuracy, but also improves the stability of generating structured outputs. Models without fine-tuning produce less stable outputs, sometimes requiring manual intervention to identify generated answers. Furthermore, their reasoning processes exhibit a higher degree of interpretability.

Table 3: Accuracy of Different LLMs; the result of the final submission is bolded

Model Name	Accuracy
ChatGLM3-6B	0.51
Mistral-7B-Instruct-v0.2	0.58
Zephyr-7B-Beta	0.56
GPT-3.5-Turbo	0.74
NumHG-ChatGLM3-6B	0.62
NumHG-zephyr-7b	0.71
NumHG-Mistral-7B	0.73
CoT-NumHG-ChatGLM3-6B	0.83
CoT-NumHG-Zephyr-7B	0.90
CoT-NumHG-Mistral-7B	**0.94**

5 Conclusion and Future Work

In this paper, we have introduced a CoT-based SFT training strategy aimed at enhancing the performance of LLMs in the task of news headline generation. Initially, we constructed the CoT-NumHG dataset, based on the existing NumHG dataset through knowledge distillation techniques. By simulating the human thought process, this dataset enhances the interpretability of the reasoning path from problem to answer. Subsequently, we utilized the CoT-NumHG dataset to perform SFT on a selected baseline model and verified significant improvements in model performance through ablation studies. The competition results further validated the efficacy of our approach, with the CoT-NumHG-Mistral-7B model achieving an accuracy rate of 94%. However, a manual review of the competition outcomes revealed some uncertainties in the model's handling of numerical information in titles, such as the need for approximations. This indicates that there is still room for improvement in understanding numerical information and generating structured outputs.

Future work will focus on the following directions: further optimization of the dataset by deduplicating and enhancing data diversity to improve the model's generalization capabilities. This includes identifying and removing duplicate or low-quality data samples, as the current proportion of copied methods is excessively high. We will adjust the proportions through sampling to address this issue. To align the generated headlines more closely with the standards of human editors, we will explore constructing a dataset in DPO (Rafailov et al., 2024) format from incorrect generation outcomes.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Sayantan Dasgupta, Trevor Cohn, and Timothy Baldwin. 2023. Cost-effective distillation of large language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7346–7354.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong, et al. 2023. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2310.19736*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, and Soujanya Poria. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

He Junqing, Pan Kunhao, Dong Xiaoqun, Song Zhuoyang, Liu Yibo, Liang Yuxin, Wang Hao, Sun Qianguo, Zhang Songxin, Xie Zejian, et al. 2023. Never lost in the middle: Improving large language models via attention strengthening question answering. *arXiv preprint arXiv:2311.09198*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization. *arXiv preprint arXiv:2203.16804*.

Kazuki Matsumaru, Sho Takase, and Naoaki Okazaki. 2020. Improving truthfulness of headline generation. *arXiv preprint arXiv:2005.00882*.

Shuyin Ouyang, Jie M Zhang, Mark Harman, and Meng Wang. 2023. Llm is like a box of chocolates: the non-determinism of chatgpt in code generation. *arXiv preprint arXiv:2308.02828*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. 2022a. Salience allocation as guidance for abstractive summarization. *arXiv preprint arXiv:2210.12330*.

Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, et al. 2023. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving

with large language models. *Advances in Neural Information Processing Systems*, 36.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

A Example of CoT-NumHG Datasets

Table 4: Example of CoT-NumHG Datasets

Input: Your task is to read and understand Article and Question. Focus on the numerical information in the article. Choose the suitable mathematical methods to arrive at the answer. The mathematical methods include copy a value from the article, trans a word into a number, paraphrase the form of digits, round a number, subtract, add, select a span from the article, divide, and multiply. Think and write three step sentences to fill in the blank in the question sentence: 1. Identifying the Relevant Information 2. Interpreting the Numerical Information 3. Choosing and Applying the Math Method Finally, output the answer in the blank. Remember, the final answer is immediately followed by Änswer:Ï #Article (Dec 25, 2020 2:25 PM) Around 1,000 British soldiers were spending Christmas Day trying to clear a huge backlog of truck drivers stuck in southeast England after France briefly closed its border to the UK, then demanded coronavirus tests from all amid fears of a new coronavirus variant. Even though an estimated 4,000 or so international truck drivers are spending yet another day cooped up in their cabs, some progress was evident Friday, with traffic around the English Channel port of Dover moving in an orderly fashion toward the extra ferries that were put on to make the short crossing across to Calais in northern France. Rail operator Eurotunnel was also back in action, the AP reports, offering a way back into France. The military personnel were directing traffic and helping a mass testing program for the drivers, who must test negative to enter France. French firefighters have been drafted to help the military test drivers for coronavirus. Poland's Territorial Defense Force also sent reinforcements to help with testing and food distribution. #Question Fill in the blank: British Troops Work to Clear ____K Trucks
Output: To fill in the blank in the question sentence: British Troops Work to Clear ____K Trucks. 1. Identifying the Relevant Information: The question asks for the number of trucks that the British troops are working to clear. 2. Interpreting the Numerical Information: The article mentions that there is an estimated 4,000 international truck drivers stuck in southeast England. 3. Choosing and Applying the Math Method: To arrive at the answer, I chose the Paraphrase method to convert the numerical information of 4,000 into a different representation. By paraphrasing 4,000 as K̈,Ï can represent the number of trucks as 4. Summary: math methods: Paraphrase(4,000,K) Answer: 4

Byun at SemEval-2024 Task 6: Text Classification on Hallucinating Text with Simple Data Augmentation

Cheolyeon Byun

byuncheolyeon@akane.waseda.jp

Abstract

This paper aims to classify sentences to see if it is hallucinating, meaning the generative language model has output text that has very little to do with the user's input, or not. This classification task is part of the Semeval 2024's task on Hallucinations and Related Observable Overgeneration Mistakes, AKA SHROOM, which aims to improve awkward-sounding texts generated by AI. This paper will first go over the first attempt at creating predictions, then show the actual scores achieved after submitting the first attempt results to Semeval, then finally go over potential improvements to be made.

1 Introduction

How AI and Large Language Models are able to understand and generate human language is elusive, to say the least. The underlying ingenuity behind the architecture of such models, involves technology and methods that are considered a black box like neural networks, named after the fact that the inner workings are impossible to grasp and properly digest for even experts (Castelvecchi, 2016). But language models are far from perfect, to the point where sometimes, text generated by complex natural language generation models are considered to be hallucinating. The term hallucination here refers to text that has been generated or processed to solve tasks like machine translation or Natural Language Generation, that are easily subject to the issue of being grammatically correct but being untethered from the user's input or the source material (Lee et al., 2019). This paper attempts to classify these hallucinating texts using different models and methods, in order to see which can get the best results in terms of accuracy.

2 Task Description

Semeval's task 6, hereinafter denoted as SHROOM, asks participants to successfully classify hallucination texts from non-hallucinating text, where each data has been annotated by 5 different annotators, where a majority vote is done to categorize each data point. Going over the JSON input data presented in Figure 1, The "hyp" row refers to the text that has been generated/processed by a model, so the output. The model here could refer to something like BERT, and can be seen as the "model" value. "src" refers to the user input or source material the model is working with in order to produce the output, and the "tgt" is the expected result that the model should be aiming for. In Figure 1, the model's "task" is "DM", or Definition Modeling. The model is expected to provide the definition for the word asked on the input. In this case input asks for the meaning of surmounting. The output of the model is "hyp", and the correct answer is "tgt". This output is put on a majority vote by 5 people, and finally the data in Figure 1 was labeled as hallucinating, where the probability of 0.6 because 3 out of 5 people voted in favor of hallucinating.

3 System Overview

3.1 Data Pipeline

The structure of the data pipeline is as follows. Language models such as BERT or RoBERTa were used to pre-process and tokenize text, which were then turned into high dimensional vectors by the word embedding layer that is able to capture rich contexts (Vaswani et al., 2017). Hyperparameters include, binary cross entropy as the loss function as it is a binary classification task, epochs were set as 20. The Adam optimizer was utilized for training the model, and learning rates were set as 0.0005. The Adam optimizer was used because it can lead to better results than stochastic gradient descent depending on the task thanks to its dynamic adjusting of learning rates,(Zhang, 2018) and tinkering around with SGD and Adam personally has led to the conclusion that Adam is slightly better in terms of accuracy.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 270–273

```
[
  {
    "hyp": "A sloping top .",
    "ref": "tgt",
    "src": "The sides of the casket were covered with heavy black broadcloth , with velvet
    caps , presenting a deep contrast to the rich surmountings . What is the meaning of
    surmounting ?",
    "tgt": "A decorative feature that sits on top of something .",
    "model": "ltg/flan-t5-definition-en-base",
    "task": "DM",
    "labels": [
      "Not Hallucination",
      "Hallucination",
      "Not Hallucination",
      "Hallucination",
      "Hallucination"
    ],
    "label": "Hallucination",
    "p(Hallucination)": 0.6
  },
```


Figure 1: SHROOM Data

3.2 Input Data

The features that were used from the input data are "src", "data", and "tgt". The three columns were concatenated into one column, but with a prefix of the column name attached at the front of the text, which resulted in a sentence like the following.

Concatenated Columns

```
"src": "<define> Infradiaphragmatic
 </define> intra- and suprasellar
 craniopharyngioma",
"tgt": "(medicine) Below the diaphragm.",
"hyp": "(anatomy) Relating to the
diaphragm."
```

This simply made working with the text data easier and while working with the validation dataset, no significant difference in accuracy was exhibited in this approach compared to using all three different columns. The idea was to let the attention mechanism of the transformer model of BERT's do most of the heavy lifting of figuring out the the context and relationship between the words, in this case the column names and the texts that follow (Vaswani et al., 2017). The column "model" was not used as it also did not lead to any change in the accuracy of all models and methods whatsoever. The text were split into train and validation datasets. Then the BERT or RoBERTa models were fine tuned so that it is able to achieve better results specifically for the classification of texts. The softmax activation function on the output layer of the neural network was used to get the predicted probability between 0 and 1 (as per Devlin et al., 2019). Besides complex models like BERT, classification methods such as logistic regression, SVC, and Naive Bayes were also used with word vectors created from BERT.

3.3 Data Augmentation

Overall, the pipeline was relatively simple, but one strategy that was employed to achieve better accuracy was to increase the amount of data available. The trial and validation data provided by Semeval was on the smaller side, which contributed to overfitting. The data also had the issue of being somewhat imbalanced with the non-hallucinating data in the validation dataset amounted to 295, whereas the hallucinating data amounting to 206. Data Augmentation was utilized to combat these issues. Data Augmentation is the modification, and augmentation of the input data itself. The text inside the input data is sometimes dropped randomly, replaced by synonyms, or words can be randomly inserted, thus creating more sentences with labels to work with. As dropping or inserting random words seemed detrimental as described by previous studies (see e.g. Wei and Zou, 2019), for this task, the data augmentation was restricted to adding data where words had been replaced by synonyms for a subset consisting of 10% of the total data.

Before Data Augmentation

```
(idiomatic, intransitive) To begin a new
endeavor with vigor.
```

After Data Augmentation

```
idiomatic intransitive to start out a new
endeavor with vigor
```

4 Results

Refer to Table 1 for the results from the initial attempt. The accuracies there are what was achieved after using a simple 80-20 train and test split on the data. Models like BERT and RoBERTa were fine-tuned while regular classification methods such as logistic regression, SVC and Naive Bayes were used with the word embedding vectors retrieved from BERT. After reviewing the results, the predictions made with BERT were submitted to Semeval, as it had the highest accuracy. However the submitted results actually achieved an accuracy of only 60%.

Table 1: Accuracy of Models/Methods

Model	Accuracy (%)
BERT	80
RoBERTa	76
SVC	50
Naive Bayes	48
Logistic Regression	46

4.1 Probability of Hallucination

A Spear-man correlation score of the expected train and test data obtained from the soft-max layer of BERT resulted in a 0.64. But the actual submission spear-man correlation coefficient was a 0.23. Compared to the top ranked team whose numbers were above 0.7, a 0.23 is a bit underwhelming, and has lots of room for improvement.

5 Conclusion and Limitations

Some potential improvements that could be employed are the following.

5.1 Cross Validation

First, cross validation instead of a simple train and test split. Though data augmentation allowed for more data, which in turn made a simple train and test split theoretically suffice, since the more data one has the less likely it is that a train test split, by pure luck, can affect the accuracy significantly, simply trying out cross validation could have led to more insight on the actual accuracy on the validation data (Bates et al., 2022).

5.2 Reduce Over-Fitting

Second, methods of reducing over-fitting. It is highly likely that BERT was being over-fit with the input data, considering how the BERT model using only the validation dataset with a train and test split resulted in a 0.8 accuracy, but a 0.6 with the final test data. To counteract over-fitting, lasso regression could be incorporated to add a penalty term for high variance (Ranstam and Cook, 2018).

5.3 Ensemble Learning

Third, the "task" column was not utilized as it did not impact accuracy in, but perhaps a different approach could have been to separate data based on tasks and then to feed those data to the pipeline. Which means a holistic ensemble learning model, that uses multiple models, whether it be using the same model or different ones, in order to get a more generalized correlation score that leads to less over-fitting can be a great method. This holistic approach can lead to not just better accuracy, but also a better spear-man correlation score. The current data pipeline did not utilize the probability feature of the input dataset, and an ensemble learning pipeline that can utilize the probability feature properly, alongside the stacking of generalization could be a way of achieving better spear-man scores. (Su et al., 2013)

5.4 Better Utilization of Data Augmentation

Fourth, a more thorough utilization of data augmentation. In supervised machine learning, limited data often leads to over-fitting, which is precisely why data augmentation was the key strategy to counter the issue, but a more thorough and systematic approach to utilizing data augmentation seems to be the key. In the final attempt, the amount of data point was a 1000 each for hallucinating text and non-hallucinating text, for a total of 2000. Perhaps starting with 2000, then 10,000, then 20,000, while trying out different strategies of how the data is augmented like random deletions and addition of words, instead of just relying on replacement of words with synonyms, would have definitely benefited this research immensely.(Ying, 2019)

6 Code

https://github.com/esohman/SemEval2024

References

Stephen Bates, Trevor Hastie, and Robert Tibshirani. 2022. Cross-validation: what does it estimate and how well does it do it?

Davide Castelvecchi. 2016. Can we open the black box of ai? *Nature*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2019. Hallucinations in neural machine translation.

J Ranstam and J A Cook. 2018. LASSO regression. *British Journal of Surgery*, 105(10):1348–1348.

Ying Su, Yong Zhang, Donghong Ji, Yibing Wang, and Hongmiao Wu. 2013. Ensemble learning for sentiment classification. In *Chinese Lexical Semantics*, pages 84–93, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China. Association for Computational Linguistics.

Xue Ying. 2019. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022.

Zijun Zhang. 2018. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–2.

DeepPavlov at SemEval-2024 Task 6: Detection of Hallucinations and Overgeneration Mistakes with an Ensemble of Transformer-based Models

Ivan Maksimov and **Vasily Konovalov** and **Andrei Glinskii**
Moscow Institute of Physics and Technology
{maksimov.ivan.v, vasily.konovalov, glinsky}@phystech.edu

Abstract

The inclination of large language models (LLMs) to produce mistaken assertions, known as hallucinations, can be problematic. These hallucinations could potentially be harmful since sporadic factual inaccuracies within the generated text might be concealed by the overall coherence of the content, making it immensely challenging for users to identify them. The goal of the SHROOM shared-task is to detect grammatically sound outputs that contain incorrect or unsupported semantic information. Although there are a lot of existing hallucination detectors in generated AI content, we found out that pretrained Natural Language Inference (NLI) models yet exhibit success in detecting hallucinations. Moreover their ensemble outperforms more complicated models.

1 Introduction

Over the past few years, Natural Language Generation (NLG) models have experienced substantial advancements, particularly due to transformer-based architectures like a Generative Pretrained Transformer (GPT) (Radford et al., 2019). However, two interconnected issues challenge the field: firstly, the tendency of present neural systems to generate incorrect yet smooth outputs and, secondly, the inadequacy of existing metrics in evaluating accuracy over fluency. This causes NLG models to "hallucinate", i.e., produce fluent but incorrect outputs that we currently struggle to detect automatically (Ji et al., 2023).

The Shared-task on Hallucinations and Related Observable Overgeneration Mistakes (SHROOM) has been suggested to address this challenge. In particular, the SHROOM task aims at addressing the existing gap in assessing the semantic correctness and meaningfulness of NLG models.[1] Within the Shared task (Mickus et al., 2024), one needs to detect grammatically sound English output that contains incorrect semantic information (i.e.,unsupported or inconsistent with the source input) in case there is no labeled training data available.

We propose to address the SHROOM task by leveraging an ensemble of pretrained transformer-based Natural Language Inference (NLI) models. The NLI models are used to derive features of hallucination probabilities, and then a tree-based gradient boosting model (Prokhorenkova et al., 2019) provides a final decision. Our results indicate that NLI-based models can be effectively used to detect hallucinations. Moreover, the ensemble model highly outperforms the base estimators in correlation with annotators' decisions.

To summarize, this work includes the following contributions:

- We conducted a systematic study, re-evaluating existing NLI models for hallucination detection tasks.
- We trained an ensemble of NLI models to detect hallucination that correlates with human judgment.

Additionally, we made the code publicly available.[2]

2 Background

Nowadays, it is well known that NLG models often generate coherent outputs that are not faithful to the given input, commonly referred as hallucinations (Maynez et al., 2020). Hallucination has been studied in a wide range of tasks, including but not limited to summarization (Huang et al., 2021), dialogue generation (Shuster et al., 2021) and a variety of other NLG tasks.

[1] https://helsinki-nlp.github.io/shroom/

[2] https://github.com/ivan-kud/semeval-2024-shroom

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 274–278

There are several benchmarks for hallucination detection. HaluEval includes 5,000 general user queries with ChatGPT responses and 30,000 task-specific examples from three tasks, i.e., question answering, knowledge-grounded dialogue, and text summarization (Li et al., 2023). FaithDial is a benchmark for hallucination-free dialogues by modifying hallucinated responses in the Wizard of Wikipedia (WoW) benchmarks (Dziri et al., 2022).

The SHROOM shared task organizers went one step further. The shared task was conducted with a newly constructed dataset of 4,000 model outputs labeled by five annotators each, including three NLP tasks: machine translation (MT), paraphrase generation (PG), and definition modeling (DM). Participants were asked to detect hallucinations in two different settings: a model-aware track where the organizers also provided a checkpoint to a model that generated the output and a model-agnostic track where they did not. The checkpoints are publicly available on HuggingFace.

All three NLG tasks are in English, with the exception of the input for the MT task, which is in Russian for the model-agnostic task and in many other languages for the model-aware task (Mickus et al., 2024).

3 Dataset

The dataset for the SHROOM challenge comprises a compilation of model-generated text entries with the aim to classify each output as either a hallucination of the generative model or not.

Information for the data sample includes the following fields: (i) *src* – the input text given to the generative language model; (ii) *hyp* – the generated textual output of the model; and (iii) *tgt* – the intended reference or the ground truth text that the model is supposed to generate; (iv) *task* – the task being solved; (v) *labels* – five labels, either *"Hallucination"* or *"Not Hallucination"* labeled by five annotators, and finally, (vi) *p(Hallucination)* indicates the proportion of annotators that labeled the data sample as a hallucination.

The dataset was split in the following way: training data of 30,000 samples without annotations with 10,000 samples for each task; validation data of 499 labeled samples with 187, 187, and 125 samples for DM, MT, and PG tasks, respectively; and test data of 1,500 examples without annotations to evaluate and rank the results of the competitors with 563, 562, and 375 examples for DM, MT, and PG tasks, respectively. Validation data sample is presented in Table 1.

All participants' submissions were evaluated using two criteria:

- Accuracy that the system reached on the binary classification.
- Spearman correlation of the system's output probabilities with the proportion of the annotators labeling the item as a hallucination.

4 Methodology

NLI task determines whether a hypothesis follows a premise and classifies it as either entailment, contradiction, or neutral. Previous research showed that NLI can be successfully used for hallucination detection in summarization (Gekhman et al., 2023).

Our system is an ensemble of several NLI pre-trained transformer-based models. Each model defines a new feature set, then a tree-based gradient boosting final estimator is trained on top of these features.

It's well known that in-domain training data are crucial for classifier performance (Konovalov et al., 2016). However, organizers do not provide us with the labeled training set, so we train final estimator on the labeled validation set.

The following is a list of NLI transformer-based models used in our pipeline:

- RoBERTa by Liu et al. (2019) is a robustly optimized BERT that outperformed BERT on the MNLI task.
- Sentence-RoBERTa by Reimers and Gurevych (2019) that returns sentence embeddings. Thus, we calculate the probability of hallucination as $1 - S$, where S is cosine similarity between the premise and hypothesis embeddings.
- DeBERTa by Manakul et al. (2023) uses an disentangled attention mechanism and an enhanced mask decoder being pre-trained on MNLI-m/mm outperformed BERT and RoBERTa of comparable sizes.
- T5 by Honovich et al. (2022) was trained on a binary problem (entailment or contradiction) on the bunch of NLU datasets: SNLI (Bowman et al., 2015), MNLI (Williams et al., 2017), Fever (Thorne et al., 2018), Scitail (Khot et al., 2018), PAWS (Zhang et al., 2019), VitaminC (Schuster et al., 2021).

Task	Input	Output	Reference	p(Halluc.)
DM	Because redpillers are usually normies or <define> Chadlites </define> while incels are ugly low - value males that women feel innately repulsed by .	(slang) An incel.	(incel, _, slang) A man of a slightly lower ranking on a scale of physical attractiveness than Chad.	0.2
MT	Ядрён батон!	Nuclear Baton!	Blimey!	1.0
PG	Are we resolved?	So, what, we're in the clear now?	Do we have a deal?	0.4

Table 1: Data examples from the validation sample of the model-agnostic track. Hallucination probability of 0.2 means that one of the annotators classified the example as a hallucination and the remaining four annotators classified it as not a hallucination. The resulting class label is determined by majority voting. For the DM task, the model had to define a word between two special tokens: <define> and </define>.

In addition to the output of the before-mentioned models, we add as features the lengths of input, output and reference texts. Then we train CatBoost (Prokhorenkova et al., 2019) models as metamodels on top of these features. Besides CatBoost model, we also train Random Forest (Breiman, 2001) implemented in scikit-learn library (Pedregosa et al., 2011) and LightGBM (Ke et al., 2017). CatBoost yields the best results among them.

5 Experimental setup

We do not use any preprocessing of input texts (premises and hypotheses). Neither do we use an unlabeled training set. So, the transformer-based models serve to obtain features from the validation and test sets, then the CatBoost metamodel is trained on the validation set and predicts the test set.

As for the CatBoost metamodel, we performed the following steps:

- We found the hyperparameters on the validation set by using Optuna (Akiba et al., 2019). Stratified k-fold cross-validation [3] with 10 splits was used for the classification model and k-fold cross-validation with 10 splits – for the regression model. The best parameters for the classification model for the model-agnostic task: `iterations = 216`, `learning_rate = 0.010`, `depth = 12`, and for the model-aware task: `iterations = 129`, `learning_rate = 0.005`, `depth = 9`. The best parameters for the regression model for the model-agnostic task: `iterations = 356`, `learning_rate = 0.029`, `depth = 5`, and for the model-aware task: `iterations = 317`, `learning_rate = 0.012`, `depth = 9`.
- We evaluated the metrics on the validation sample using repeated stratified k-fold cross-validation with 10 splits and 5 repeats.
- We trained it on the whole labeled validation set.
- We predicted test set labels.

6 Results

The results on the test set for both model-agnostic and model-aware tracks are presented in Table 2. There are scores for the baseline provided by organizers, best scores from the leader-board, individual transformer-based models and our system as a whole.

Among NLI pre-trained models, T5 model significantly outperformed other NLI models. However, our ensemble approach using features from all NLI pre-trained models significantly outperformed T5 in terms of correlation with annotators' decisions.

Our approach for model-agnostic case provided us with an accuracy of 82.1% and Spearman correlation of 0.752. With this approach, our team achieved the 6th place out of 41 in the competition for model-agnostic track. Only two teams achieved a higher Spearman correlation.

The same approach was applied for the model-aware track and provided us with an accuracy of 79.9%, which is the 8th place out of 38 in the com-

[3] https://en.wikipedia.org/wiki/Cross-validation_(statistics)

Model	model-agnostic		model-aware	
	Accuracy	**Corr.**	**Accuracy**	**Corr.**
`nli-roberta-large`	62.8	0.608	66.2	0.566
`roberta-large-mnli`	73.7	0.611	73.0	0.549
`deberta-base-mnli`	72.8	0.617	73.1	0.597
`deberta-large-mnli`	75.7	0.701	75.5	0.688
`deberta-xlarge-mnli`	73.5	0.699	74.4	0.681
`deberta-v2-xlarge-mnli`	74.4	0.711	74.7	0.677
`deberta-v2-xxlarge-mnli`	76.1	0.729	75.9	0.691
`deberta-selfchecknli`	75.3	0.683	75.9	0.683
`t5_xxl_true_nli_mixture`	81.1	0.650	79.6	0.626
`baseline`	69.7	0.403	74.5	0.488
Our system$_{\text{submitted}}$	82.1	0.752	79.9	0.713
Our system$_{\text{best}}$	82.5	0.757	79.9	**0.722**
Best leaderboard	**84.7**	**0.770**	**81.3**	0.715

Table 2: The results of the accuracy and Spearman correlation metrics on the test sample for the model-agnostic and model-aware tracks.

petition. The value of Spearman correlation turned out to be 0.713.

More detailed results of the competition can be found in Mickus et al. (2024).

7 Conclusion

In this paper, we describe the ensemble system for hallucination detection by using transformer-based models. We present a simple, yet effective ensemble pipeline that provided us with results comparable with the best scores for the both tracks.

Future work might include thoughtful error analysis. Improved quality can be achieved by annotating unlabeled training set with LLMs (Ostyakova et al., 2023). In addition, a multilingual setup of NLI models can be used to develop multilingual hallucination detection system (Chizhikova et al., 2023; Konovalov et al., 2020). The proposed approach can be used standalone or can be integrated into the DeepPavlov framework (Burtsev et al., 2018).

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference.

L. Breiman. 2001. Random forests. *Machine Learning*, 45:5–32.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, and Vasily Konovalov. 2018. Deeppavlov: An open source library for conversational ai. In *NIPS*.

Anastasia Chizhikova, Vasily Konovalov, and Mikhail Burtsev. 2023. Multilingual case-insensitive named entity recognition. In *Advances in Neural Computation, Machine Learning, and Cognitive Research VI*, pages 448–454, Cham. Springer International Publishing.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M. Ponti, and Siva Reddy. 2022. Faithdial: A faithful benchmark for information-seeking dialogue.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023. Trueteacher: Learning factual consistency evaluation with large language models. *arXiv preprint arXiv:2305.11171*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Vasily Konovalov, Pavel Gulyaev, Alexey Sorokin, Yury Kuratov, and Mikhail Burtsev. 2020. Exploring the bert cross-lingual transfer for reading comprehension. In *Computational Linguistics and Intellectual Technologies*, pages 445–453.

Vasily Konovalov, Oren Melamud, Ron Artstein, and Ido Dagan. 2016. Collecting Better Training Data using Biased Agent Policies in Negotiation Dialogues. In *Proceedings of WOCHAT, the Second Workshop on Chatbots and Conversational Agent Technologies*, Los Angeles. Zerotype.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Lidiia Ostyakova, Veronika Smilga, Kseniia Petukhova, Maria Molchanova, and Daniel Kornev. 2023. ChatGPT vs. crowdsourcing vs. experts: Annotating open-domain conversations with speech functions. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 242–254, Prague, Czechia. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2019. Catboost: unbiased boosting with categorical features.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

HIJLI_JU at SemEval-2024 Task 7: Enhancing Quantitative Question Answering Using Fine-tuned BERT Models

Partha Sarathi Sengupta
Computer Science and Engineering
Jadavpur University, Kolkata
jitendriyo@gmail.com

Sandip Sarkar
Computer Science and Application
Hijli College, Kharagpur
sandipsarkar.ju@gmail.com

Dipankar Das
Computer Science and Engineering, Jadavpur University, Kolkata
dipankar.dipnil2005@gmail.com

Abstract

In data and numerical analysis, Quantitative Question Answering (QQA) becomes a crucial instrument that provides deep insights for analyzing large datasets and helps make well-informed decisions in industries such as finance, healthcare, and business. This paper explores the "HIJLI_JU" team's involvement in NumEval Task 1 within SemEval 2024, with a particular emphasis on quantitative comprehension. Specifically, our method addresses numerical complexities by fine-tuning a BERT model for sophisticated multiple-choice question answering, leveraging the Hugging Face ecosystem. The effectiveness of our QQA model is assessed using a variety of metrics, with an emphasis on the f1_score() of the scikit-learn library. Thorough analysis of the macro-F1, micro-F1, weighted-F1, average, and binary-F1 scores yields detailed insights into the model's performance in a range of question formats.

1 Introduction

Quantitative Question Answering (QQA) is a crucial tool in the large field of data and numerical analysis because it uses sophisticated computer methods to extract and interpret significant information from large datasets. Imagine it as a strong force that has particular sway over important industries like finance, healthcare, and business, where it plays a crucial role in forecasting trends and assisting in the making of well-informed decisions. QQA acts as a driving force behind wise decisions by skillfully converting apparently complicated data into useful knowledge and clearing a way through the complexities of numerical data.

Our team, "HIJLI_JU," participated in Task 1 (Quantitative Understanding) of NumEval within SemEval 2024[1], thereby actively engaging in the competitive landscape. An annual series of international natural language processing (NLP) competitions called SemEval (Semantic Evaluation) evaluates the state-of-the-art in a range of tasks pertaining to semantic analysis and understanding. These challenges serve as a forum for practitioners and scholars from both academia and industry to investigate and expand the field of computational linguistics. SemEval is well known for its broad range of tasks, which address a variety of difficulties in natural language processing.

The presented QQA task in the context of the SemEval 2024 NumEval competition provides a platform for researchers and developers to showcase advancements in quantitative understanding (num). SemEval, an annual challenge, encompasses diverse language tasks, including sentiment analysis and word meanings, contributing to the ongoing progress in systems designed for language understanding and processing. Table 1 shows the example of the SemEval-2024 Task dataset.

This paper is organized as follows: in Section 2, a survey of related literature is presented, and in Section 3, a detailed description of the dataset is provided. Section 4 explores the details of our suggested model while Section 5 explains the experimental setup. Experiments using our model are shown in Section 6, and observations are discussed in Section 7. Bringing everything together, we wrap up the paper in Section 8 and offer some suggestions for future directions for study.

2 Related Work

The HIJLI_JU team participated in the IJCNLP-2017 Task 5 on Multi-choice Question Answering, focusing on vector representations and machine learning for classification (Sarkar et al., 2017). Their model, designed exclusively for English language questions. The methodology involves representing questions and answers in vector space, computing cosine similarity, and employing a classification approach to identify the correct answer

[1] https://sites.google.com/view/numeval/tasks

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 279–284

Task	Question	Answer
QP	FED'S DUDLEY REPEATS EXPECTS GDP GROWTH TO PICK UP IN 2014, FROM [Masked] PCT POST-RECESSION AVERAGE	1
QNLI	S1: Nifty traded above 7500, Trading Calls Today, S2: Nifty above 7400	Entailment
QQA	Elliot weighs 180 pounds whereas Leon weighs 120 pounds. Who has a bigger gravity pull? Option1: Elliot Option2: Leon	Option 1

Table 1: Task Questions and Answers

option.

Sandip presented a novel approach to enhance science-based Multiple Choice Question Answering (MCQA) systems by leveraging distributed semantic similarity and a classification approach. Three models (Model 1, Model 2, and Model 3) were developed to address differences in dataset formats, specifically focusing on IJCNLP Task 5 and SciQ datasets (Sarkar et al., 2020).

Zuccon looks into using fancy word techniques from computer language models for finding information better. They use special methods to understand words and put them into a translation model. The results show that this approach improves how well information is found, and it's flexible – it works well even if the word understanding is done differently or comes from a different set of information (Zuccon et al., 2015).

Researchers in the field of Quantitative Question Answering (QQA) have been investigating ways to enhance computer systems' capacity to respond to numerical questions. They've tried a range of tactics, such as deep learning and sophisticated machine learning. To improve response accuracy, certain studies might incorporate external data.

For QQA, standardized tests (datasets) covering a range of numerical questions in disciplines like science and finance are being developed. They're also developing equitable methods to evaluate the efficacy of these Q&A platforms.

The brittleness of existing AI systems, including large-scale language models, in arithmetic reasoning within natural language understanding is addressed by the proposed multi-task benchmark NUMGLUE (Mishra et al., 2022a). In order to prepare AI systems for increasingly difficult mathematical tasks, the benchmark attempts to promote the development of systems that are able to reason robustly about arithmetic in language.

EQUATE is a framework assessing quantitative reasoning in textual entailment for natural language understanding systems (Ravichander et al., 2019). State-of-the-art models don't consistently outperform a basic baseline, highlighting a potential gap in implicit quantity reasoning. This framework aims to spur the development of models focusing on quantitative reasoning in language understanding.

Chen and his colleague investigates whether neural network models can acquire numeracy skills, focusing on predicting numeral magnitudes in text (Chen et al., 2019). Introducing the Numeracy-600K benchmark dataset, the study explores various models. Additionally, they highlights a practical application scenario by demonstrating the task's utility in detecting exaggerated information.

Chen also addresses innumeracy issues in pretrained language models, focusing on the fundamental task of teaching language models to understand numerals in text (Chen et al., 2023). It suggests a method that combines a comparing-number task with number notation exploration, modification, and pre-finetuning. Their research shows enhanced performance in three benchmark datasets for tasks related to quantitative analysis, especially for RoBERTa.

question	Option1	Option2	answer	type
Jame's mother has a photo of Jane standing at a height of 14 inches, whereas a mountain appears to have height of 26 cm. It looks that way because?	the mountain was farther away	Jane was farther away	Option 2	Type_3
Tina is racing her two dogs. Her greyhound weighs 40 kgs, and her rottweiler weighs 35 kgs. The dog that gets faster more quickly is the?	rottweiler	greyhound	Option 1	Type_3
A toddler is rolling a ball for more than 1 mins on the grass and rolls it on to the sand where it stops after 43 seconds. The sand stopped the ball because it has _____ than the grass.?	more friction	less friction	Option 1	Type_3
The fish glided with a speed of 4 mph through the water and 1 mph through the jello because the _____ is smoother.?	jello	water	Option 2	Type_3

Table 2: Example of SemEval-2024 Task 7 Dataset

3 Dataset

Semeval-2024 Utilizing current benchmark datasets for three different task types—Quantitative Prediction (QP), Quantitative Natural Language Inference (QNLI), and Quantitative Question Answering (QQA)—is the one of the task of NumEval Task 1: Quantitative Understanding (Chen et al., 2023; Mishra et al., 2022b). Managing numbers, forecasting numerical values, deciphering logical connections in numerical sentences, and responding to inquiries requiring numerical data are all part of these tasks. The goal is to assess and improve the performance of models in handling these quantitative tasks.

The provided data format appears to represent a set of questions along with options, correct answers, and additional attributes [2], [3]. Table 2 shows the different fields of Task 1 of NumEval Dataset. On the other hand, Table 3 gives the description of the statics of the dataset. Here's a description of the key components in the data format:

- **"question":** The primary question text is contained in this field.
- **"Option1" and "Option2":** There are two options available to answer the question in these fields.
- **"answer":** Indicates which option is the correct answer.
- **"type":** Specifies the type of question.
- **"question_sci_10E":** Represents the same question as "question" but with numerical values expressed in scientific notation (e.g., 1.4000000000×10^1 inches).
- **"question_char":** Represents the same question with numerical values written as characters (e.g., "1 4 inches").
- **"question_sci_10E_char":** Combines scientific notation and characters for numerical values in the question.
- **"question_mask":** Presents the question with placeholders like "[Num]" indicating where numerical values are expected to be filled.

In summary, this data format is designed to offer various question formats, including options and the

[2] https://drive.google.com/drive/folders/10uQI2BZrtzaUejtdqNU9Sp1h0H9zhLUE?usp=sharing

[3] https://sites.google.com/view/numeval/data

right response, as well as various ways to represent numerical values (scientific notation, characters, masked placeholders, etc.). It appears to be intended as a test of numerical information interpretation and comprehension in various formats.

4 System Description

We explored the realms of natural language processing, immersing ourselves in Hugging Face's dynamic ecosystem known for its transformative libraries such as transformers and datasets. At the heart of our machine learning endeavor was the fine-tuning of a BERT model for nuanced multiple-choice question answering, including numerical complexities. Hugging Face's BertTokenizer and TFBertForMultipleChoice powered our training, and the fine-tuned model effortlessly transitioned into competent inference on the test dataset [4].

We started our data preparation process by adding "context" and "label" to JSON files. Next, we transformed the data into Dataset objects that were kept in a DatasetDict. We managed a preprocess_function, used BertTokenizer, and used the Datasets map method with 'batched=True' with caution to optimize our operations. Performance was improved by enabling dynamic sentence padding during collation using the Data Collator For Multiple Choice modification. Figure 1 shows the system description of HIJLI_JU for the participation in SemEval-2024 Task 7.

BERT, or Transformers' Bidirectional Encoder Representations, which had been pre-trained using masked language modeling and next sentence predictions on a substantial amount of unlabeled text data. Our approach was based on its bidirectional capabilities, which enabled it to simultaneously capture semantic subtleties from both sides.

5 Experimental Setup

We set up the parameters for training with a batch size of 16 over ten epochs, a starting learning rate of 0.00001, and no warm-up phases. Our datasets' dimensions, comprising 564 examples for training, 81 for development, and 162 for testing, demonstrated accuracy. The purpose of this rigorous training setup was to guarantee our QQA model's generalizability and robustness.

Using Jupyter Notebooks, Google Colab is a cloud-based platform that offers an interactive and collaborative environment for Python coding. Well-known for providing free access to GPU and TPU resources, Colab has grown to be a well-liked option in the data science and machine learning domains. Its smooth integration with Google Drive makes sharing and collaborative editing simple, which improves the effectiveness of team projects. Because of its intuitive interface and free availability of robust computational resources, Google Colab is an indispensable resource for individuals and groups working on a variety of computational tasks.

6 Results

In our quest to evaluate the efficacy of our Quantitative Question Answering (QQA) model, we employed a comprehensive set of metrics and examined its performance across various question formats. The `scikit-learn` library's `f1_score()` function served as our tool for this evaluation, offering insights into the model's proficiency in different contexts.

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Here, TP represents the number of true positives, FP represents the number of false positives, and FN represents the number of false negatives.

The F1 Score is a fundamental metric in machine learning, providing a balanced evaluation of classification models by combining precision and recall. This versatile metric has several variants, each suited to different scenarios. In this essay, we delve into macro-F1, micro-F1, weighted-F1, average, and binary-F1, exploring their applications and significance.

6.1 Macro-F1 Score

The macro-F1 Score calculates the F1 Score for each class independently and then computes the unweighted average. This approach treats all classes equally, making it valuable when assessing a model's performance across diverse classes without bias towards larger ones.

[4] https://huggingface.co/docs/transformers/en/tasks/multiple_choice

Files	Size
QQA_train.json	564
QQA_dev.json	81
QQA_test.json	162

Table 3: Statistics of SemEval-2024 Task 7 Dataset

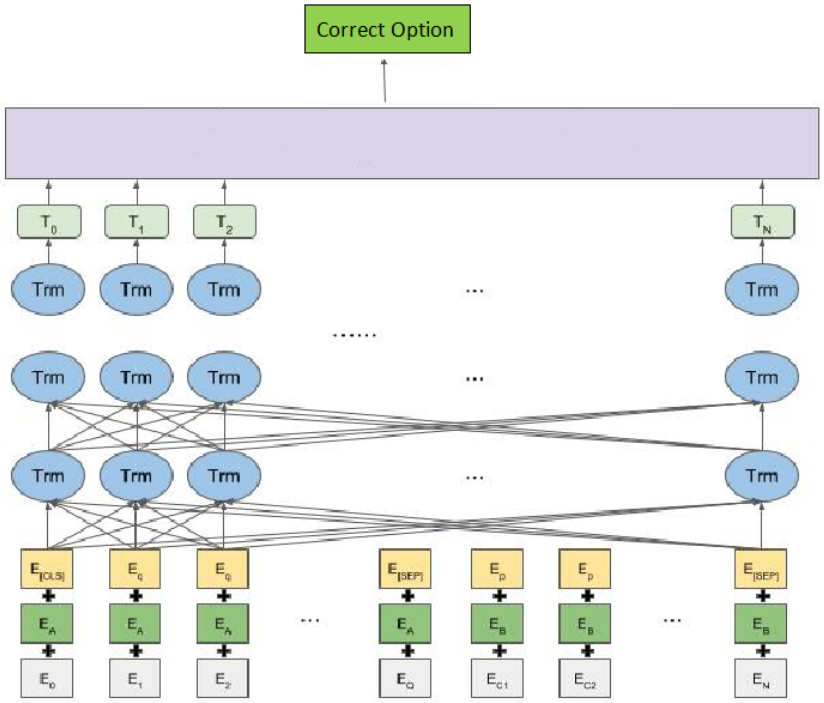


Figure 1: BERT Model

6.2 Micro-F1 Score

In contrast, the micro-F1 Score aggregates the contributions of all classes into a single F1 Score. Particularly useful for imbalanced datasets, it considers the varying sizes of different classes, providing an overall evaluation that accounts for class imbalances.

6.3 Weighted-F1 Score

The weighted-F1 Score extends the macro-F1 approach by considering class sizes. It calculates F1 Scores for each class and then computes a weighted average based on the number of instances in each class. This adjustment ensures that larger classes contribute proportionally more to the overall score.

6.4 Average F1 Score

The term "average F1 Score" is a general descriptor that encompasses various approaches to aggregating F1 Scores across multiple classes. It may refer to micro-F1, macro-F1, or other weighted or unweighted averages, depending on the context.

6.5 Binary F1 Score

The binary F1 Score is the traditional F1 Score applied to a binary classification problem with two classes – positive and negative.

7 Observations

The observed results highlight the nuanced performance of the Quantitative Question Answering (QQA) model across different question formats. Notably, questions presented in the character format consistently outperform other representations, demonstrating its robustness in handling diverse classes independently, particularly in imbalanced datasets. The Macro-F1, Micro-F1, and Weighted-F1 scores consistently identify the question_char format as the most effective in achieving a balanced evaluation. This format excels not only in independently handling varied classes but also in proportionally contributing to overall performance based on class instances. The Average F1 scores further affirm the versatility of the question_char format, emphasizing its capacity for a well-rounded

Field used	Macro-F1	Micro-F1	Weighted F1	average=None	Binary F1
question	0.50344	0.50617	0.50345	array([0.46667, 0.54023])	0.54023
question_char	0.53058	0.53704	0.53058	array([0.47552, 0.58564])	0.58564
question_sci_10E	0.44026	0.44444	0.44026	array([0.48864, 0.39189])	0.39189
question_sci_10E_char	0.51489	0.51852	0.51489	array([0.47297, 0.55682])	0.55682

Table 4: Result of HIJLI_JU on SemEval-2024 Task 7

evaluation across multiple classes.

8 Conclusion and Future Work

In conclusion, we found that Quantitative Question Answering (QQA) is like a helpful tool for understanding numbers better. It's useful in important areas like business, healthcare, and finance, helping with predicting trends and making smart decisions. QQA is like a guide that empowers organizations and people to understand and use tricky data. The research we did shows how important QQA is for understanding numbers better.

Looking ahead, there are exciting possibilities for more research on QQA. We could explore new tasks and find ways to use QQA in specific areas like healthcare. Working with experts in different fields could help make QQA more useful in different situations. Also, we can improve how we measure the success of QQA and make it better by using the latest technology and techniques in language understanding. This ongoing exploration will keep pushing QQA to new places and make it even more important in understanding both language and numbers.

References

Semeval-2024 task 7: Numeral-aware language understanding and generation.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022a. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022b. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Sandip Sarkar, Dipankar Das, and Partha Pakray. 2017. JU NITM at IJCNLP-2017 task 5: A classification approach for answer selection in multi-choice question answering system. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 213–216, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Sandip Sarkar, Dipankar Das, Partha Pakray, and David Eduardo Pinto Avendaño. 2020. Developing MCQA framework for basic science subjects using distributed similarity model and classification based approaches. *Int. J. Asian Lang. Process.*, 30(3):2050015:1–2050015:18.

Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. 2015. Integrating and Evaluating Neural Word Embeddings in Information Retrieval. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, pages 12:1–12:8, New York, NY, USA. ACM.

NCL Team at SemEval-2024 Task 3: Fusing Multimodal Pre-training Embeddings for Emotion Cause Prediction in Conversations

Shu Li[1] and **Zicen Liao**[2] and **Huizhi Liang**[3]
[1] Beijing Accent Advertising Co., Ltd.
[2] School of Computing, Newcastle University, Newcastle upon Tyne, UK
[3] School of Computing, Newcastle University, Newcastle upon Tyne, UK
15510366636@163.com
and liaozicen55@gmail.com
and huizhi.liang@newcastle.ac.uk

Abstract

In this study, we introduce an MLP approach for extracting multimodal cause utterances in conversations, utilizing the multimodal conversational emotion causes from the ECF dataset. Our research focuses on evaluating a bi-modal framework that integrates video and audio embeddings to analyze emotional expressions within dialogues. The core of our methodology involves the extraction of embeddings from pre-trained models for each modality, followed by their concatenation and subsequent classification via an MLP network. We compared the accuracy performances across different modality combinations including text-audio-video, video-audio, and audio only.

1 Introduction

In recent times, multimodal sentiment analysis has become a critical research frontier in the realm of natural language processing, moving beyond the confines of traditional text analysis to embrace a richer blend of audio, visual, and text data. This comprehensive approach aims to deepen our understanding of sentiments and emotions.

Previous research has highlighted the effectiveness of hierarchical fusion techniques and context modelling in improving the precision of multimodal sentiment analysis by adeptly merging features from varied modalities (Wang et al., 2023). Additionally, initiatives such as the Unified Multimodal Sentiment Analysis and Emotion Recognition UniMSE have proven the benefits of applying contrastive learning techniques to enhance performance in both sentiment analysis and emotion recognition, underscoring the significance of integrated frameworks within this field (Hu et al., 2022). CubeMLP delves into the realm of feature mixing for multimodal data processing (Sun et al., 2022). Meanwhile, the MMLatch model sheds light on the critical roles of bottom-up and top-down fusion mechanisms (Paraskevopoulos et al., 2022), offering insights into the impact of high-level representations on the synthesis of sensory information.

This study proposes the development of a Multilayer Perceptron network, specifically designed to extract causal utterances from the Multimodal Emotion-Cause Pair Extraction in Conversations (ECF) dataset (Wang et al., 2024).

2 Data Description

For this research, the ECF dataset has been selected as the primary source of data for training and testing our model. The ECF dataset contains several key elements that are integral to our study:

- Video Clips: Each sample in the dataset includes a video clip from the show Friends, capturing the visual expressions, body language, and interactions between characters.
- Audio Tracks: Audio tracks in the video clips, which include the spoken dialogues, tone of voice, laughter, and other paralinguistic features.
- Transcribed Text: For each clip, the spoken dialogues are transcribed to provide textual context to the interactions.
- Emotion and Sentiment Annotations: The dataset provides detailed annotations for each dialogue segment, including the emotion category, the emotion utterance, and the cause utterance.

Our research leverages the video and audio components of the ECF dataset. By analyzing the video and audio modalities, our goal is to uncover the underlying patterns and triggers of emotional expressions, without the direct influence of textual information.

To adapt the ECF dataset for our specific research objectives, a meticulous data preparation

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 285–290

process is undertaken. This involves: - Annotation Mapping: Aligning the cause utterance annotations with the corresponding audio and video segments for supervised learning. - Dataset Split: The dataset is divided into training validating subsets as 8:2, the testing subset is provided by the task provider.

Our research endeavours to architect a model that harnesses the strengths of each modality to provide a comprehensive understanding of sentiment. At the core of our methodology is a model architecture designed to seamlessly integrate these diverse data types, leveraging the power of pre-trained models to extract embeddings from text, video, and audio streams for sentiment extraction and classification.

The model lies in the process of concatenating the embeddings generated by these modularity extractors. This approach not only preserves the richness of each modality data but also facilitates the creation of a unified representation that embodies the composite sentiment conveyed across text, video, and audio. The concatenated embeddings serve as input to a Multilayer Perceptron (MLP) classifier, which is designed to discern the integrated sentiment.

3 Methodology

We propose an MLP network architecture designed to synergize the embeddings extracted from video and audio. This network aims to process and integrate these multimodal inputs, facilitating the classification of cause utterances within the framework of sentiment analysis without relying on textual information. The decision to exclude textual data from our analysis stems from a desire to investigate the intrinsic value of audio-visual cues in sentiment analysis.

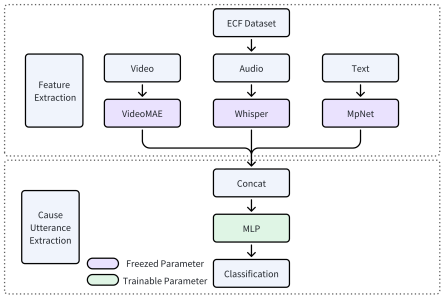


Figure 1: Overview of the cause utterance classification model.

As shown in Figure 1, this model contains two parts, feature extraction and classification. First, we extract audio, video, and text embedding from the pre-trained model. The text embedding extraction is for the comparison of experiments. Then the embeddings are concatenated and put into the MLP network which acts as the classifier for extracting cause utterances. To facilitate the extraction of emotion category and cause utterance, these tasks are regarded as classification tasks. The labels associated with the emotion category and cause utterance are regarded as the classes for these two tasks. This approach enables the MLP to execute the classification.

3.1 Embedding Extraction

VideoMAE is utilized for extracting video embeddings. This model, based on the Masked Autoencoder principle, selectively masks portions of the input video frames and reconstructs the missing parts, thereby learning robust video representations. (Tong et al., 2022) Given an input video V, the model produces an embedding E_V as follows:

$$E_V = \text{VideoMAE}(V) \tag{1}$$

To obtain the video embedding, frames are initially extracted from the video at their native resolution and compiled into a list. Temporal subsampling is applied to this collection of frames, a measure aimed at reducing computational time. Each subsampled set of frames applied normalization and resizing as data augmentation before being inputted into the pre-trained VideoMAE model to acquire the corresponding embeddings.

Whisper is used to extract audio embeddings from the corresponding audio tracks. Whisper processes the raw audio signals, focusing on capturing the nuances of speech, tone, and other auditory features relevant to sentiment analysis (Radford et al., 2022). For an audio input A, the Whisper model outputs an embedding E_A as:

$$E_A = \text{Whisper}(A) \tag{2}$$

Audio information was segregated from the video content and resampled to a 16000Hz sample rate to align with the Whisper model. In leveraging the pre-trained Whisper model for embedding extraction, the classification head was removed to get the pooler output.

MPNet is used to extract text embeddings. MPNet integrates the strengths of both masked language modelling (MLM) and permuted language modelling (PLM) to effectively capture the context

of words in a sentence, both from left-to-right and right-to-left, making it effective for understanding the full context of textual data.For an text input T, the Whisper model outputs an embedding E_T as:

$$E_T = \text{MPNet}(T) \tag{3}$$

3.2 Integration of Embeddings

The embeddings E_V and E_A are concatenated to form a unified representation E_{VA} of the video and audio modalities:

$$E_{VA} = \text{Concat}(E_V, E_A) \tag{4}$$

The embeddings E_V, E_A and E_T are concatenated for the ablation test:

$$E_{VAT} = \text{Concat}(E_V, E_A, E_T) \tag{5}$$

This concatenated embedding serves as the input to the MLP network. The decision to concatenate these embeddings is based on the hypothesis that doing so preserves the distinctiveness of each modality while allowing the network to learn from the intermodal dynamics, essential for identifying cause utterances.

3.3 Network Design

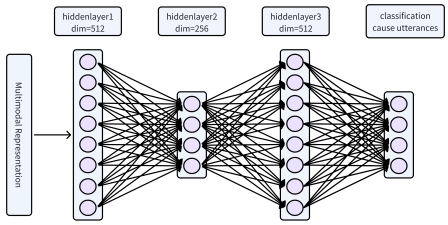


Figure 2: An MLP model aims to classify cause utterance from multimodal embeddings.

Our model employs a MLP architecture, crafted to process and classify concatenated video and audio embeddings. The simplicity and interpretability were significant considerations in choosing the MLP network as the classifier.

The MLP consists of four fully connected layers. The first layer expands the input to 512 hidden units, followed by a reduction to 256 units in the second layer, and an expansion back to 512 units in the third layer, before concluding with the output layer that matches the number of cause utterance classes.

Each hidden layer is equipped with a ReLU activation function to introduce non-linearity, allowing the model to learn complex patterns in the data. To combat overfitting, a dropout rate of 0.5 is applied after each ReLU activation, regularizing the network by randomly omitting a subset of features at each iteration of the training process.

The MLP network is designed with four fully connected layers, integrating nonlinear activation functions and dropout for regularization. Given the concatenated embedding E_{VA}, the forward pass through the MLP can be described by the following set of equations:

- **First Layer Transformation:** The input is passed through the first fully connected layer, transforming it to a higher-dimensional space.

  $$H_1 = \text{ReLU}(W_1 E_{VA} + b_1) \tag{6}$$

  where W_1 and b_1 are the weights and biases of the first linear layer, respectively, and E_{VA} represents the concatenated embeddings. ReLU activation follows to introduce non-linearity.

- **Applying Dropout:** To prevent overfitting, dropout is applied to the output of the ReLU activation,

  $$D_1 = \text{Dropout}(H_1) \tag{7}$$

- **Second and Third Layer Transformations:** The second and third layers further process the data through linear transformations and ReLU activations:

  $$H_2 = \text{ReLU}(W_2 D_1 + b_2) \tag{8}$$

  and

  $$H_3 = \text{ReLU}(W_3 D_2 + b_3) \tag{9}$$

  where W_2, W_3, b_2, and b_3 correspond to the weights and biases of these layers. Each transformation is followed by dropout to enhance model generalization.

- **Final Layer Transformation:** The last step in the network involves passing the output through a final fully connected layer without subsequent ReLU activation, resulting in the output logits,

  $$O = W_4 D_3 + b_4 \tag{10}$$

  This layer maps the processed features to the target output space.

Where W_i and b_i represent the weights and biases of the i^{th} layer, respectively, and ReLU is the Rectified Linear Unit activation function. The dropout is applied after each activation except the final layer to mitigate overfitting.

The output O represents the logits corresponding to each class, which in this case are the possible cause utterances. The model employs a cross-entropy loss function to compute the difference between the predicted probabilities and the actual class labels. This loss guides the training process through backpropagation, adjusting the weights W_i and biases b_i to minimize prediction errors. The network is optimized using the Adam optimizer, with a learning rate of 0.0002.

4 Experiments

In the subtask2 dataset, 1,374 conversations have been annotated by human evaluators. The dataset comprises 13,619 video clips, each tagged with a cause utterance label, delineating the specific cause associated with the clip. These cause utterances are distributed across 29 distinct categories. 66.49 percentage of the cause utterances can be attributed to the context provided by the current video clip itself.

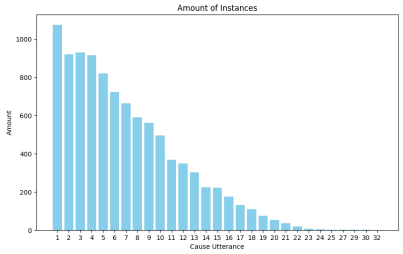


Figure 3: Overview of the cause utterance instances distribution. The cause utterance tends to be more related to earlier situations.

The histogram of Fig.3 illustrates the frequency of instances for each cause utterance category, with a descending order of occurrence. Categories are indexed from 1 to 32 on the x-axis, reflecting a diverse range of causes utterances. The y-axis quantifies the amount of instances, highlighting the prevalence of lower-indexed categories.

4.1 Training Process

The training of our MLP model follows a systematic approach. We utilize the cross-entropy loss, which combines a softmax activation and a log loss in one function. This choice is particularly suited for multi-class classification problems, as it measures the performance of a classification model whose output is a probability value between 0 and 1. The Adam optimizer is chosen for its effectiveness in handling sparse gradients and adapting the learning rate for each parameter, which is crucial given the complexity of our model and the diverse nature of our data. The learning rate is set to 0.0002, offering a balance between fast convergence and the risk of overshooting minimal loss. Our model undergoes training for 2000 epochs. This training period ensures that the model has the opportunity to learn from the entire dataset, optimizing its parameters to identify cause utterances. The alignment of these choices with our research objectives and dataset characteristics ensures a rigorous yet efficient training process, tailored to maximize performance while mitigating the risk of overfitting.

4.2 Metrics

To evaluate the model's performance, we employ the F1 score and weighted F1 score as our primary metrics. These metrics are particularly chosen for their relevance in classification tasks.

F1 Score is calculated as the harmonic mean of precision (P) and recall (R), providing a comprehensive measure of the model's accuracy across all classes. It is given by the equation:

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{11}$$

This metric effectively balances the precision and recall, offering a singular view of model performance.

Weighted F1 Score extends the F1 score by weighting each class's score according to its presence in the dataset. This adjustment makes the metric more representative of the model's performance across classes of varying sizes. The weighted F1 score can be expressed as:

$$\text{Weighted F1} = \sum_{i=1}^{n} w_i \times F1_i \tag{12}$$

where w_i is the weight or relative frequency of class i in the dataset, and $F1_i$ is the F1 score for class i. This calculation ensures the final score reflects the proportional significance of each class, making it invaluable for datasets with class imbalances. These metrics provide an assessment of the model's performance, reflecting its effectiveness in classifying the embeddings in alignment with cause utterance extraction.

5 Ablation Studies

In the context of investigating the ECF dataset, our research undertook a series of ablation studies. These studies were aimed at elucidating the impact of various combinations of modalities on the efficacy of cause utterance classification. These studies are crucial for understanding how combining video, audio, and text data can enhance performance. These studies also help people assess the individual impact of each modality of data on the task cause utterance classification. Ablation Study Design The ablation studies were designed to compare the following configurations:
- Utilization of video, audio, and text embeddings. We utilized video, audio, and text embeddings to assess the maximum potential of multimodal data fusion. This configuration represents the most comprehensive approach.
- Utilization of video and audio embeddings without text. By employing video and audio embeddings while excluding text, our objective is to test whether the information conveyed by the audio modality is equivalent to that of the text modality. This comparison helps us understand the extent to which visual and auditory information alone can drive the classification process.
- Utilization of either video or audio embeddings exclusively. This test helps determine the standalone capabilities of visual and auditory data in identifying cause utterances.

5.1 Ablation Study Results

The network design does not incorporate any combination of modalities.

Configuration	F1	wF1
Video + Audio + Text	0.0253	0.0552
Video + Audio	0.0237	0.0694
Video Only	0.0144	0.0255

Table 1: Ablation Study Results on the ECF Dataset development set.

Configuration	F1	wF1
Video + Audio + Text	-	-
Video + Audio	0.0152	0.0146
Video Only	0.0222	0.0119

Table 2: Ablation Study Results on the ECF Dataset test set.

The combination of video and audio embeddings emerged as a configuration, showcasing its ineffectiveness in the absence of textual data.

6 Conclusion

We proposed a bimodal framework incorporating visual and acoustic modalities for emotion extraction from the "Friends" series, with the addition of a text modality to discern its performance enhancement. The results demonstrate that as the number of modalities increases, the accuracy of emotion extraction gradually improves. Particularly, the Visual-Acoustic model exhibits relatively good accuracy, with a significant improvement upon the addition of the textual modality. The experiment highlights:

- The crucial role of the text modality in sentiment analysis.
- In scenarios lacking textual data, the application of bi-modal models incorporating visual and acoustic modalities can effectively accomplish recognition tasks.

However, the experiment has several limitations concerning the target task. For instance, it did not utilize state-of-the-art pre-trained models, resulting in intra-modality comparisons without specifying the most suitable model for the task. To overcome this limitation, we will develop an evaluation system in our future work to further investigate the effects of embedding extraction using different modalities with pre-trained models. Due to time and resource constraints, the experiment did not extensively tune the models, thereby might not show their optimal performance. Future research could explore using Multi-modal LLMs and task-specific pre-trained models to predict emotion cause in conversations.

References

Guimin Hu, Ting-En Lin, Yi Zhao, Guangming Lu, Yuchuan Wu, and Yongbin Li. 2022. Unimse: Towards unified multimodal sentiment analysis and emotion recognition.

Georgios Paraskevopoulos, Efthymios Georgiou, and Alexandros Potamianos. 2022. Mmlatch: Bottom-up top-down fusion for multimodal sentiment analysis.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Hao Sun, Hongyi Wang, Jiaqing Liu, Yen-Wei Chen, and Lanfen Lin. 2022. Cubemlp: An mlp-based

model for multimodal sentiment analysis and depression estimation. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22. ACM.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

DeBERTa at SemEval-2024 Task 9: Using DeBERTa for Defying Common Sense

Marco Siino
Department of Electrical, Electronic
and Computer Engineering
University of Catania
Italy
marco.siino@unipa.it

Abstract

The significant achievements of language models have motivated researchers in the natural language processing (NLP) community to confront challenges requiring nuanced and implicit reasoning, inspired by human-like common-sense understanding. Although efforts focusing on vertical thinking tasks have received substantial recognition, there remains a notable lack of investigation into lateral thinking puzzles. To bridge this void, the authors at SemEval-2024 propose BRAINTEASER: a multiple-choice Question Answering task designed meticulously to assess the model's lateral thinking capabilities and its capacity to question default common-sense assumptions. Specifically, at the SemEval-2024 Task 9, for the first subtask (i.e., Sentence Puzzle) the organizers asked the participants to develop models able to reply to multi-answer brain-teasing questions. For this purpose, we propose the application of a DeBERTa model in a zero-shot configuration. The proposed approach achieves an aggregate score of 0.250. Suggesting a significant room for improvements in future works.

1 Introduction

Human reasoning encompasses two fundamental types of cognitive processing: vertical and lateral thinking. Vertical thinking is marked by its sequential and analytical approach, drawing upon principles of rationality, logic, and rule-following, often attributed to the left-brain hemisphere (Knauff, 2013; Huang et al., 2023). This mode of thinking is essential for creating logical pathways, such as understanding physical scenarios or solving riddles based on direct associations. In contrast, lateral thinking, often referred to as "thinking outside the box," is a creative cognitive process. It entails exploring problems from unconventional perspectives and challenging ingrained assumptions. Lateral thinking, associated with the right-brain hemisphere, is crucial for resolving unconventional puzzles by defying common-sense associations and considering alternative perspectives.

While natural language processing (NLP) models have made significant strides in vertical thinking tasks, particularly in the field of large language models (LLMs). Their performance in lateral thinking remains largely unexplored. LLMs have demonstrated remarkable performance across various reasoning tasks, even when provided with minimal or no training examples. These models excel in tasks requiring vertical thinking abilities, such as reasoning over physical interactions and social implications (Siino et al., 2022b), showcasing strong common-sense association and inference capabilities. However, prior research has largely overlooked the evaluation of LLMs' lateral thinking abilities, as creative thinking problems are often filtered out during data preprocessing, and only those aligned with common-sense associations are retained.

To address this gap, a novel benchmark called BRAINTEASER (Jiang et al., 2023) to evaluate the lateral thinking abilities of state-of-the-art LLMs is proposed at SemEval-2024 Task 9 (Jiang et al., 2024). The organizers frame lateral thinking puzzles as multiple-choice Question Answering (QA) tasks, a format that is intuitive for humans to engage with and straightforward to assess automatically. The BRAINTEASER benchmark comprises two tasks: Sentence Puzzles and Word Puzzles, designed to assess lateral thinking at different levels of granularity. To develop the dataset, organizers employ a data collection pipeline that retrieves relevant puzzles from publicly available websites, filters out irrelevant question categories, and ensures high data quality. Additionally, to mitigate concerns regarding LLM memorization and consistency, the organizers enhance BRAINTEASER with two reconstruction strategies: semantic reconstruction and context reconstruction. These strate-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 291–297

gies aim to promote deeper understanding and reasoning rather than mere memorization of patterns.

To meet these objectives, there is a growing demand for automated tools capable of understanding data using recent advancements in NLP models. The emergence of machine and deep learning architectures has sparked increased interest in NLP, prompting substantial efforts to develop techniques for automated identification and understanding of textual content available on the internet. In the literature, various strategies have been proposed so far. Over the past fifteen years, some of the most successful approaches have included Support Vector Machines (SVM) (Colas and Brazdil, 2006; Croce et al., 2022), Convolutional Neural Networks (CNN) (Kim, 2014; Siino et al., 2021), Graph Neural Networks (GNN) (Lomonaco et al., 2022), ensemble models (Miri et al., 2022; Siino et al., 2022), and Transformers (Vaswani et al., 2017).

The sections of this paper are structured as follows: Section 2 offers background information on Task 9, held at SemEval-2024. In Section 3, we outline the approach introduced in this study. Section 4 delves into the specifics of the experimental setup employed to reproduce our findings. The outcomes of the official task and relevant discussions are presented in Section 5. Finally, Section 6 concludes our study and suggests avenues for future research.

We make all the code publicly available and reusable on GitHub[1].

2 Background

The increasing adoption of Transformer-based architectures in academic research has also been bolstered by various methodologies showcased at SemEval 2024. These methodologies tackle diverse tasks and yield noteworthy findings. For instance, at the Task 2 (Jullien et al., 2024), where to address the challenge of identifying the inference relation between a plain language statement and Clinical Trial Reports is used T5 (Siino, 2024c); Task 4 (Dimitrov et al., 2024) where is employed a Mistral 7B model to detect persuasion techniques in memes (Siino, 2024b); and Task 8 (Wang et al., 2024), that utilizes a DistilBERT model to identify machine-generated text (Siino, 2024a).

The Task 9 hosted at SemEval-2024, is based on the human reasoning processes comprising the two already-mentioned types of thinking: vertical and lateral.

Specifically, the BRAINTEASER QA task consists of two subtasks: the Sentence and the Word Puzzle ones, that require awareness of commonsense "defaults" and overwriting them through unconventional thinking that distinguishes these defaults from hard constraints.

In detail, for the Sentence Puzzle one, the puzzle defying common-sense is centred on sentence snippets. On the other hand, for the Word Puzzle subtask, the response diverges from the conventional interpretation of the word and concentrates on the letter arrangement within the target question.

Both subtasks incorporate an adversarial subset, crafted by manually altering the original brainteasers while preserving their underlying reasoning paths.

An example from the official CodaLab page[2] takes as example the following original sentence:

> *"A man shaves everyday, yet keeps his beard long."*

The four possible explanations are:

1. **He is a barber.**
2. He wants to maintain his appearance.
3. He wants his girlfriend to buy him a razor.
4. None of the above.

However, the task organizers also included two other samples based on the previous one. In these two cases, a semantic and a contextual reconstruction have been made to challenge a classification model. The two reconstructions (with the same four possible explanations as in the original) are:

- SEMANTIC RECONSTRUCTION: *"A man preserves a lengthy beard despite shaving every day."*
- CONTEXT RECONSTRUCTION: *"Tom attends class every day but doesn't do any homework."*

3 System Overview

Despite evidence suggesting that Transformers may not always yield optimal results for every text classification task (Siino et al., 2022a), various strategies, such as domain-specific fine-tuning (Sun et al.,

[1] https://github.com/marco-siino/SemEval2024/

[2] https://codalab.lisn.upsaclay.fr/competitions/15566

2019; Van Thin et al., 2023) and data augmentation (Lomonaco et al., 2023; Mangione et al., 2022; Siino et al., 2024a), have proven to be advantageous depending on the task's objectives.

However, to address the task 9 hosted at SemEval-2024 we made use of a zero-shot learning approach (Chen et al., 2023; Wahidur et al., 2024), making use of the DeBERTa Transformer (He et al., 2020).

Our approach is zero-shot (Pourpanah et al., 2022) and make use of the above-mentioned DeBERTa model. Specifically, we employed the multilingual version 3 fine-tuned on the SQuAD2.0 dataset[3]. DeBERTa improves upon the BERT and RoBERTa models by introducing disentangled attention mechanisms and an enhanced mask decoder. Leveraging these enhancements, DeBERTa outperforms RoBERTa across most Natural Language Understanding (NLU) tasks when trained on a dataset of 80GB in size. In DeBERTa V3, efficiency is further enhanced by integrating ELECTRA-Style pre-training with Gradient Disentangled Embedding Sharing. Comparative analysis against DeBERTa reveals notable enhancements in model performance across downstream tasks in the V3 version. Further elaboration on the novel techniques employed in this new model can be found in the original paper. The version of DeBERTa utilized in our experiments is mDeBERTa, a multilingual variant of DeBERTa. It maintains an identical architecture while being trained on CC100 multilingual data. The mDeBERTa V3 base model comprises 12 layers with a hidden size of 768. It encompasses 86 million backbone parameters and a vocabulary of 250,000 tokens, resulting in 190 million parameters in the embedding layer. This model underwent training using 2.5 trillion tokens of CC100 data, akin to the XLM-R model.

For the experimental settings, we started evaluating several prompt engineering strategies (White et al., 2023; Liu et al., 2023) to optimize the model replies and to obtain satisfactory results guided by the labelled samples in the training set. For example, we included in the prompt/question to the model, the premise that the given question is a brain-teaser one. Furthermore, we also evaluated the performance of the model on the training set using a few-shot learning setup. In this case, we provided as input (included in the prompt) ten questions indicating the correct answer. Also in this case we did not obtain satisfactory results.

More specifically, given the task hosted at SemEval-2024, we asked the model: *"What is the correct answer to the brain teaser question from the following choices? (Pick only one Option (A)-(D)"*. To this request, the model replied with one or more words that we parsed to extract one of the choices. For example, given the context:

> *"Romeo and Juliet are discovered dead on the bedroom floor. Glass shards and some water were on the floor when they were found. A bookcase and a bed are the sole pieces of furniture in the space. Other than the neighboring railroad track, the house is located in a rural area. How is that even doable? "*

And our question:

> *"What is the correct answer to the brain teaser question from the following choices? (Pick only one Option (A)-(D)"*

And the answers/options:

> (A): They were sleeping and scared by the sound of track.
>
> (B): The rumble of the train moved the shelf which crushed them.
>
> (C): Romeo and Juliet are fish. The rumble of the train knocked the tank off the shelf, it broke and Romeo and Juliet did not survive.
>
> (D): None of above.

The model replied with:

> *"Romeo and Juliet are fish."*

that we mapped into the label *2* corresponding to the third answer. Finally, we collected all the predictions provided on the test set to into a JSON file with required format to submit our predictions.

During our experiments to build our prompt, we also evaluated other LLMs like GPT-Neo and GPT-NeoX (Gao et al., 2020). However, on the labelled training set, we found better performance of DeBERTa in the responses provided. It is also worth notice that we conducted several experiments to find an effective prompt strategy to address the task.

As indicated in a recent investigation by Siino et al. (Siino et al., 2024b), preprocessing does

[3] https://rajpurkar.github.io/SQuAD-explorer/

not significantly impact text classification tasks when employing Transformers. Specifically, the optimal combination of preprocessing strategies closely resembles the performance achieved without any preprocessing at all, particularly in the context of Transformer models. Therefore, to maintain a highly efficient and computationally lightweight system, we opted not to apply any preprocessing to the text.

4 Experimental Setup

We implemented our model on Google Colab. The library we used come from Hugging Face and as already mentioned is a multilingual version of DeBERTa[4]. The dataset provided for all the phases are available on the official competition page. We did not perform any additional fine-tuning on the model. To run the experiment, a T4 GPU from Google has been used. After the generation of the predictions, we exported the results on the format required by the organizers. As already mentioned, all of our code is available on GitHub.

5 Results

Participants in Brain Teaser may participate in any or all of the two subtasks. The organizers created two adversarial questions, semantic and context reconstruction, for each brain-teaser (examples can be found on the Task Home Page). The evaluation metrics applied are Instance-based Accuracy and Group-based Accuracy, defined as follows:

- Instance-based Accuracy: Each question, whether original or adversarial, is treated as an individual instance. Accuracy is reported for the original question, its semantic reconstruction, and context reconstruction.

- Group-based Accuracy: Questions and their corresponding adversarial instances are grouped together. A system earns a score of 1 only if it correctly answers all questions within the group. Accuracy is reported for original and semantic reconstruction and original and semantic and context reconstruction.

In Table 1, we present the outcomes derived from our methodology. They are the same results publicly available on the official final ranking shown on the official task page[5]. The results are about the sentence task, given the fact that we did not take part in the word-related task.

DeBERTa	
Original	0.225
Semantic	0.250
Context	0.275
Ori+Sem	0.200
Ori+Sem+Con	0.075
Overall	0.250

Table 1: The method's performance on the test set. In the table are reported the results obtained and shown on the official task page.

Table 2 presents the performance results of the top three teams alongside the results achieved by the final-ranking team, as displayed on the official task page. While our straightforward approach shows potential for enhancement compared to the top-performing models, it is noteworthy that our method required no additional pre-training. Moreover, the computational resources needed to address the task were manageable, utilizing the free online resources provided by Google Colab.

6 Conclusion

This paper introduces the utilization of a DeBERTa model for addressing Task 9 at SemEval-2024. In our submission, we opted for a zero-shot learning approach, leveraging a pre-trained and fine-tuned Transformer model without further adaptation. Through various experiments, we found it advantageous to construct a prompt containing the question for the model. Subsequently, we provided the context, question, and answer candidates as the prompt, prompting the model to discern the correct candidate answer. Despite the task's inherent complexity, as evidenced by the final ranking, there remains ample room for improvement.

Potential alternative methodologies include leveraging the few-shot learning capabilities of the model or exploring alternative models such as GPT and T5. Additionally, integrating additional data or incorporating samples from training and development sets could yield performance enhancements. Further refinements could be achieved through fine-tuning and framing the problem as a text classification task. Moreover, given the promising results

[4] https://huggingface.co/timpal0l/mdeberta-v3-base-squad2

[5] https://codalab.lisn.upsaclay.fr/competitions/15566

TEAM NAME	Original	Semantic	Context	Ori+Sem	Ori+Sem+Con	Overall
abdelhak (1)	1.000	1.000	0.950	1.000	0.950	0.983
lulu13gjdfnglgr (2)	1.000	0.975	0.925	0.975	0.900	0.967
Maxine (3)	0.975	0.975	0.925	0.975	0.900	0.958
wwangbw (31)	0.300	0.175	0.150	0.075	0.025	0.208

Table 2: Comparing performance on the test set. In the table are shown the results obtained by the first three teams and by the last one. In parentheses is reported the position in the official final ranking.

observed across various tasks, the adoption of few-shot learning or data augmentation strategies could also be explored for improved outcomes (Wang et al., 2023; Maia et al., 2024; Siino et al., 2023; Meng et al., 2024; Muftie and Haris, 2023; Tapia-Téllez and Escalante, 2020; Siino and Tinnirello, 2023).

While our straightforward approach demonstrates potential for refinement, it is noteworthy that it required no additional pre-training. Moreover, the computational resources needed to address the task were manageable, utilizing the free online resources provided by Google Colab.

Acknowledgments

We extend our gratitude to the anonymous reviewers for their insightful comments and valuable suggestions, which have significantly enhanced the clarity and presentation of this paper.

References

Shiming Chen, Ziming Hong, Wenjin Hou, Guo-Sen Xie, Yibing Song, Jian Zhao, Xinge You, Shuicheng Yan, and Ling Shao. 2023. Transzero++: Cross attribute-guided transformer for zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12844 – 12861.

Fabrice Colas and Pavel Brazdil. 2006. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Daniele Croce, Domenico Garlisi, and Marco Siino. 2022. An SVM ensemble approach to detect irony and stereotype spreaders on twitter. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2426–2432. CEUR-WS.org.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, and Hai-Tao Zheng. 2023. Lateval: An interactive llms evaluation benchmark with incomplete information from lateral thinking puzzles. *arXiv preprint arXiv:2308.10855*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Markus Knauff. 2013. *Space to reason: A spatial theory of human thought*. Mit Press.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. 2023. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Francesco Lomonaco, Gregor Donabauer, and Marco Siino. 2022. COURAGE at checkthat!-2022: Harmful tweet detection using graph neural networks and ELECTRA. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 573–583. CEUR-WS.org.

Francesco Lomonaco, Marco Siino, and Maurizio Tesconi. 2023. Text enrichment with japanese language to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2708–2716. CEUR-WS.org.

Beatriz Matias Santana Maia, Maria Clara Falcão Ribeiro de Assis, Leandro Muniz de Lima, Matheus Becali Rocha, Humberto Giuri Calente, Maria Luiza Armini Correa, Danielle Resende Camisasca, and Renato Antonio Krohling. 2024. Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer. *Expert Systems with Applications*, 241:122418.

Stefano Mangione, Marco Siino, and Giovanni Garbo. 2022. Improving irony and stereotype spreaders detection using data augmentation and convolutional neural network. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2585–2593. CEUR-WS.org.

Zong Meng, Zhaohui Zhang, Yang Guan, Jimeng Li, Lixiao Cao, Meng Zhu, Jingjing Fan, and Fengjie Fan. 2024. A hierarchical transformer-based adaptive metric and joint-learning network for few-shot rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(3).

Mohsen Miri, Mohammad Bagher Dowlatshahi, Amin Hashemi, Marjan Kuchaki Rafsanjani, Brij B Gupta, and W Alhalabi. 2022. Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*, 37(12):11319–11341.

Fuad Muftie and Muhammad Haris. 2023. Indobert based data augmentation for indonesian text classification. In *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, page 128 – 132.

Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. 2022. A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*.

Marco Siino. 2024a. Badrock at SemEval-2024 Task 8: DistilBERT to Detect Multigenerator, Multidomain and Multilingual Black-Box Machine-Generated Text. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024b. Mcrock at semeval-2024 task 4: Mistral 7b for multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024c. T5-medical at semeval-2024 task 2: Using t5 medical embeddings for natural language inference on clinical trial data. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2021. Detection of hate speech spreaders using convolutional neural networks. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 2126–2136. CEUR-WS.org.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2022a. Fake news spreaders detection: Sometimes attention is not all you need. *Information*, 13(9):426.

Marco Siino, Marco La Cascia, and Ilenia Tinnirello. 2022b. Mcrock at semeval-2022 task 4: Patronizing and condescending language detection using multi-channel cnn, hybrid lstm, distilbert and xlnet. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 409–417. Association for Computational Linguistics.

Marco Siino, Francesco Lomonaco, and Paolo Rosso. 2024a. Backtranslate what you are saying and i will tell who you are. *Expert Systems*, n/a(n/a):e13568.

Marco Siino, Maurizio Tesconi, and Ilenia Tinnirello. 2023. Profiling cryptocurrency influencers with few-shot learning using data augmentation and ELECTRA. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2772–2781. CEUR-WS.org.

Marco Siino and Ilenia Tinnirello. 2023. Xlnet with data augmentation to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2763–2771. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2022. T100: A modern classic ensemble to profile irony and stereotype spreaders. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2666–2674. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024b. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

José Medardo Tapia-Téllez and Hugo Jair Escalante. 2020. Data augmentation with transformers for text classification. In *Advances in Computational Intelligence*, pages 247–259, Cham. Springer International Publishing.

Dang Van Thin, Duong Ngoc Hao, and Ngan Luu-Thuy Nguyen. 2023. Vietnamese sentiment analysis: An overview and comparative study of fine-tuning pretrained language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Rahman S. M. Wahidur, Ishmam Tashdeed, Manjit Kaur, and Heung-No Lee. 2024. Enhancing zero-shot crypto sentiment with fine-tuned language model and prompt engineering. *IEEE Access*, 12:10146 – 10159.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Trans. Circuits Syst. Video Technol.*, 33(12):7789–7802.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

TransMistral at SemEval-2024 Task 10: Using Mistral 7B for Emotion Discovery and Reasoning its Flip in Conversation

Marco Siino
Department of Electrical, Electronic
and Computer Engineering
University of Catania
Italy
marco.siino@unipa.it

Abstract

The EDiReF shared task at SemEval 2024 comprises three subtasks: Emotion Recognition in Conversation (ERC) in Hindi-English code-mixed conversations, Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations, and EFR in English conversations. The objectives for the ERC and EFR tasks are defined as follows: 1) Emotion Recognition in Conversation (ERC): In this task, participants are tasked with assigning an emotion to each utterance within a dialogue from a predefined set of possible emotions. The goal is to accurately recognize and label the emotions expressed in the conversation; 2) Emotion Flip Reasoning (EFR): This task involves identifying the trigger utterance(s) for an emotion-flip within a multi-party conversation dialogue. Participants are required to pinpoint the specific utterance(s) that serve as catalysts for a change in emotion during the conversation. In this paper we only address the first subtask (ERC) making use of an online translation strategy followed by the application of a Mistral 7B model together with a few-shot prompt strategy. Our approach obtains an F1 of 0.36, eventually exhibiting further room for improvements.

1 Introduction

Affective computing has experienced a resurgence, largely propelled by recent advancements in artificial intelligence. Emotion Recognition in Conversations (ERC) has emerged as a prominent task within affective computing, garnering increasing attention (Poria et al., 2019; Kumar et al., 2023). Its objective is to discern the emotion conveyed in each utterance during conversations, with implications for various applications including the development of effective dialogue systems, facilitating social viewpoint mining, and creating intelligent medical systems. Current research in ERC primarily focuses on capturing the emotional state of speakers through contextual analysis and establishing distinct contexts for different speakers, often leveraging multimodal data to support this endeavour. Despite recent strides, two major challenges persist: (1) Ensuring emotional consistency and (2) Generating contextual information. Current research efforts broadly fall into two categories: the first involves obtaining contextual representations of utterances using temporal neural networks, while the second entails capturing long-distance information through graph networks. However, these approaches overlook a crucial aspect: changes in utterance order can alter the meaning of utterances, potentially leading to varying emotional expressions. A shift in utterance order impacts the underlying meaning of the utterance, consequently influencing the speakers' emotions.

The increasing demand for automated tools capable of extracting and categorizing data from online sources underscores the need to address both established and emerging societal concerns efficiently. Recent strides in machine and deep learning architectures have sparked significant interest in Natural Language Processing (NLP). Efforts have been intensified towards developing techniques for automating the identification and categorization of textual content prevalent on the internet today. In the literature, various strategies have been proposed for performing text classification tasks. Over the past fifteen years, some of the most successful strategies have included Support Vector Machines (SVM) (Colas and Brazdil, 2006; Croce et al., 2022), Convolutional Neural Networks (CNN) (Kim, 2014; Siino et al., 2021), Graph Neural Networks (GNN) (Lomonaco et al., 2022), ensemble models (Miri et al., 2022; Siino et al., 2022), and more recently, Transformers (Vaswani et al., 2017; Siino et al., 2022b).

At SemEval-2024 Task 10 (Kumar et al., 2024a) – Emotion Discovery and Reasoning its Flip in Conversation (EDiReF) – three Subtasks were proposed. All the three subtasks are presented and

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 298–304

discussed in the Section 2.

To face with the first subtask (ERC), we proposed a Transformer-based approach which made use of Mistral 7B (Jiang et al., 2023). We used the model in a particular few-shot way described in the rest of this paper. Specifically, after translating all the code-mixed samples in English, we provided the samples from the labelled train and dev set to the model, asking while prompting to predict the emotion for the current utterance (i.e., the current sample from the test set).

The rest of the paper is made as follows. In Section 2 we provide some background on the Task 10 hosted at SemEval-2024. In Section 3 we provide a description of the approach presented. In Section 4 we provide details about the experimental setup to replicate our work. In Section 5, the results of the official task and some discussions are provided. In section 6 we present our conclusion and proposals for future works.

We make all the code publicly available and reusable on GitHub[1].

2 Background

This section furnishes background information regarding Task 10 (Subtask 1), held at SemEval-2024.

The EDiReF shared task at SemEval 2024 (Kumar et al., 2024a) encompasses three distinct subtasks, namely: Emotion Recognition in Conversation (ERC) within Hindi-English code-mixed conversations, Emotion Flip Reasoning (EFR) within Hindi-English code-mixed conversations, and EFR within English conversations (Kumar et al., 2022, 2024b). The ERC task involves the assignment of emotions to each utterance within a dialogue, drawn from a predefined set of potential emotions. Conversely, the EFR task focuses on identifying the trigger utterance(s) responsible for inducing an emotion-flip within a multi-party conversation dialogue.

In the Figure 1, is reported a sample from the official competition website[2] and specifically related to the Subtask 1 (i.e., ERC).

For Subtask 1, a submission entails a singular JSON file with each emotion in a new line. Every emotion correspond to each utterance in the official provided test set.

The second and the third subtasks differ on the language used. In one case, the language is code-mixed Hindi-English and in another case only in English. In both cases, the participants were asked to propose automatic detection systems able to detect a trigger utterance that determined a changed in the emotion. Also in this case, an example from the official task webpage is reported in the Figure 2. The EFR sample contains a trigger (i.e., 1) in proximity of the fourth sentence contained in the dialogue.

Speaker	Utterance
Sp_1	Aaj to bhot awful day tha! *(I had an awful day today!)*
Sp_2	Oh no! Kya hua? *(Oh no! What happened?)*
Sp_1	Kisi ne mera sandwich kha liya! *(Somebody ate my sandwich!)*
Sp_2	Me abhi tumhare liye new bana deti hun! *(I can make you a new one right now!)*
Sp_1	Wo great hoga! Thanks! *(That would be great! Thanks!)*

ERC aims to assign emotions to each utterance

Speaker	Utterance	Emotion
Sp_1	Aaj to bhot awful day tha! *(I had an awful day today!)*	Sad
Sp_2	Oh no! Kya hua? *(Oh no! What happened?)*	Sad
Sp_1	Kisi ne mera sandwich kha liya! *(Somebody ate my sandwich!)*	Sad
Sp_2	Me abhi tumhare liye new bana deti hun! *(I can make you a new one right now!)*	Joy
Sp_1	Wo great hoga! Thanks! *(That would be great! Thanks!)*	Joy

The example dialogue has two emotion flips. Sp_1's emotion changed from Sad to Joy while Sp_2's emotion also shifted from Sad to joy. EFR aims to justify such emotion flips using triggers.

Figure 1: Example of some samples from the dataset. In this case, we report samples related to the first ERC task in which we took part.

Speaker	Utterance	Emotion	Trigger
Sp_1	Aaj to bhot awful day tha! *(I had an awful day today!)*	Sad	0
Sp_2	Oh no! Kya hua? *(Oh no! What happened?)*	Sad	0
Sp_1	Kisi ne mera sandwich kha liya! *(Somebody ate my sandwich!)*	Sad	0
Sp_2	Me abhi tumhare liye new bana deti hun! *(I can make you a new one right now!)*	Joy	1
Sp_1	Wo great hoga! Thanks! *(That would be great! Thanks!)*	Joy	0

Figure 2: Example of some samples from the dataset. In this case, we report samples related to the EFR task.

3 System Overview

While it has been established that Transformers may not always be the optimal choice for text classification tasks (Siino et al., 2022a), the efficacy of various strategies, such as domain-specific fine-tuning (Sun et al., 2019; Van Thin et al., 2023) and data augmentation (Lomonaco et al., 2023; Mangione et al., 2022), depends on the specific objectives.

The increasing adoption of Transformer-based architectures in academic research has also been bolstered by various methodologies showcased at SemEval 2024. These methodologies tackle di-

[1] https://github.com/marco-siino/SemEval2024/
[2] https://lcs2.in/SemEval2024-EDiReF/

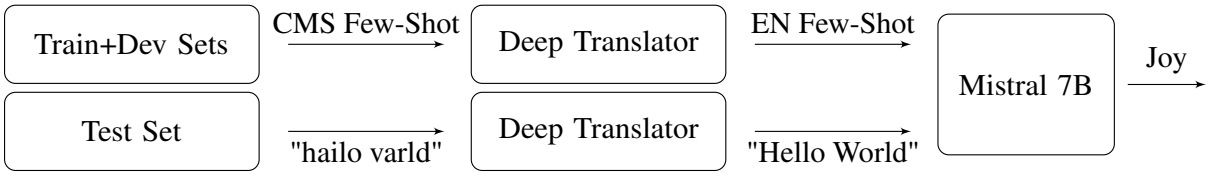


Figure 3: The system overview of our proposed approach. Given a set of Code Mixed Samples (i.e., CMS) from the train and dev sets, they are translated to English using Deep Translator (i.e., ENglish samples). Then they are all provided as input - a few-shot samples from the training set, together with the emotion definitions - to Mistral 7B. Following these few shot samples and the definitions as input, there is one utterance from the test set for which the emotion has to be predicted.

verse tasks and yield noteworthy findings. For instance, at the Task 2 (Jullien et al., 2024), where to address the challenge of identifying the inference relation between a plain language statement and Clinical Trial Reports is used T5 (Siino, 2024c); Task 4 (Dimitrov et al., 2024) where is employed a Mistral 7B model to detect persuasion techniques in memes (Siino, 2024b); and Task 8 (Wang et al., 2024), that utilizes a DistilBERT model to identify machine-generated text (Siino, 2024a).

Our approach is few-shot (Littenberg-Tobias et al., 2022) and make use of Mistral 7B. Mistral 7B, a language model boasting 7 billion parameters, is engineered to excel in both performance and efficiency. In comparison to the leading open 13B model (Llama 2), Mistral 7B demonstrates superior performance across all assessed benchmarks. Furthermore, it surpasses the top released 34B model (Llama 1) in tasks related to reasoning, mathematics, and code generation. The model capitalizes on grouped-query attention (GQA) to expedite inference, complemented by sliding window attention (SWA) to effectively process sequences of varying lengths while minimizing inference costs. Additionally, a fine-tuned variant, Mistral 7B – Instruct, is tailored for adhering to instructions, and it outperforms Llama 2 13B – chat model across both human and automated benchmarks. The introduction of Mistral 7B Instruct underscores the ease with which the base model can be fine-tuned to achieve notable performance enhancements.

For our task, before prompting the model with the current sample from the test set, we made an online and real-time use of *Google Translator* from the *deep_translator*[3] library. Then we randomly selected eighty samples from the provided labelled training set and other eighty from the provided labelled dev set. Then we formatted the samples in each set in the following way:

[3] https://pypi.org/project/deep-translator/

EMOTIONS
1. *disgust*
2. *joy*
3. *neutral*
4. *anger*
5. *sadness*
6. *contempt*
7. *surprise*
8. *fear*

Table 1: The list of all the motions available for the task.

speaker1 - utterance1 - emotion1
speaker2 - utterance2 - emotion2
...
speakerX - utterance80 - emotionY

After merging the formatted samples from both the training and the dev set, we fed the model, appending to the few-shot samples the current unlabelled sample from the official test set. At this point, the full text containing the few-shot samples plus the sample to be classified were provided as prompts to Mistral.

Then the question provided as prompt to the model was: *" Use the CONTEXT to complete the SENTENCE using ONLY one emotion among: disgust, joy, neutral, anger, sadness, contempt, surprise, fear. Do not explain!"*. Where the *CONTEXT* were the few-shot samples provided. For all the samples from the test set the model correctly predicted one of the available emotions from the list provided and shown in the Table 1.

As noted in the recent study by (Siino et al., 2024b), the contribution of preprocessing for text classification tasks is generally not impactful when using Transformers. More specifically, the best combination of preprocessing strategies is not very different from doing no preprocessing at all in the case of Transformers. For these reasons, and to keep our system fast and computationally light, we

have not performed any preprocessing on the text.

4 Experimental Setup

Our model implementation was executed on Google Colab, utilizing the Mistral 7B library from Hugging Face, specifically the Mistral-7B-Instruct-v0.2-GGUF [4] version. Additionally, we utilized the *deep_translator* package with Google Translator [5] for the translation task. The Mistral 7B version employed represents an enhanced iteration of the Mistral-7B-Instruct-v0.1 model, geared towards instruction fine-tuning. Instructions for instruction fine-tuning should be enclosed within [INST] and [/INST] tokens, with the initial instruction beginning with a sentence identifier, and subsequent instructions omitting this identifier. The generation process is terminated by the end-of-sentence token ID. Furthermore, we imported the Llama library (Touvron et al., 2023) from *llama_cpp*, with comprehensive details available on GitHub [6].

All datasets required for the various phases of the experiment are accessible on the Official Competition page. No additional fine-tuning was conducted on the model. The experiment was executed using a T4 GPU provided by Google. Upon generating the predictions, the results were exported in the format specified by the organizers. As previously mentioned, our complete codebase is accessible on GitHub.

5 Results

As described on the official task page[7], the evaluation criteria for the three tasks are delineated as follows:

- Task 1 (ERC for code-mixed): Weighted F1 score computed across all emotions.
- Task 2 (EFR for code-mixed): F1 score computed specifically for triggers (label '1.0').
- Task 3 (EFR for English): F1 score computed for triggers (label '1.0') in English.

To generate the prediction file:

1. For Task 1: Each line should depict an emotion associated with an utterance, with no additional lines separating dialogues. Each emotion should be in lowercase, devoid of extra spaces.
2. For Tasks 2 and 3: Each line should represent either 0.0 or 1.0, reflecting the label of triggers assigned to an utterance. The formatting should adhere to a string format with a floating precision of 1 (e.g., 0.0 or 1.0, rather than 0 or 1).

Then, it was asked to aggregate the outputs from all tasks into a single file named 'answer.txt', structured as follows:

- Lines 1–1580: Predictions for Task 1.
- Lines 1581–9270: Predictions for Task 2.
- Lines 9271–17912: Predictions for Task 3.

Upon compilation, the organizers asked to create a zip file encompassing *'answer.txt'* and proceed with submission as per guidelines.

	T1-F1	T2-F1	T3-F1
TransMistral 7B	0.36	0.10	0.22

Table 2: The method's performance on the test set. Even if we did not participate in the tasks 2 and 3, in the *answer* file we included a list of 0s to complete all the lines as suggested by the task organizers.

In the Table 2 we report the result obtained by the proposed approach on the official test set. Thanks to our application of an online translation followed by a Mistral 7B we have been able to reach the twenty-second position on the final ranking for the evaluation phase.

The Table 3 presents the performance outcomes achieved by the top three teams and the last-ranked team, as delineated on the official task page. While our simplistic approach showcases potential for enhancement in comparison to the leading models, it is noteworthy that our method necessitated no additional pre-training. Moreover, the computational resources utilized to tackle the task remained feasible, courtesy of the complimentary resources provided by Google Colab.

[4] https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF
[5] https://pypi.org/project/deep-translator/
[6] https://github.com/ggerganov/llama.cpp
[7] https://codalab.lisn.upsaclay.fr/competitions/16769

TEAM NAME	T1-F1	T2-F1	T3-F1
MasonTigers	0.78 (1)	0.79 (2)	0.79 (1)
Knowdee	0.73 (2)	0.66 (4)	0.61 (9)
IASBS	0.70 (3)	0.12 (7)	0.25 (12)
GAVx	0.08 (34)	0.79 (2)	0.76(2)

Table 3: Comparing performance on the test set. In the table are shown the results obtained by the first three users and by the last one ordered considering the first task. In parentheses is reported the position for each task in the official final ranking.

6 Conclusion

This paper presents the application of Mistral 7B-model for addressing the Task 10 at SemEval-2024. For our submission, we decided to follow few-shot learning approach, employing as-is, an in-domain pre-trained Transformer. After several experiments, we found it beneficial to build a prompt containing some samples from the training and from the dev set. Then we provide as a prompt the current translated sample together with the few-shot samples. The model was asked to select one of the emotion among the ones available. The task presents inherent challenges, with evident scope for refinement, as underscored by the final ranking. Potential alternative methodologies encompass leveraging the zero-shot capabilities inherent in other models such as GPT and T5, expanding the training set size through the incorporation of additional data, or adopting alternative strategies for integrating ontology-based domain knowledge beyond the approaches delineated in our study. Furthermore, refinement opportunities exist through fine-tuning and recontextualizing the problem as a text classification task.

Furthermore, given the interesting results recently provided on a plethora of tasks, also other few-shot learning (Wang et al., 2023; Maia et al., 2024; Siino et al., 2023; Meng et al., 2024) or data augmentation strategies (Muftie and Haris, 2023; Siino et al., 2024a; Tapia-Téllez and Escalante, 2020; Siino and Tinnirello, 2023) could be employed to improve the results. Looking at the final ranking, our simple approach exhibits some room for improvements. However, it is worth notice that it required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab. Also, thanks to the proposed approach, we have been able to outperform the baseline provided by the task organizers.

Acknowledgments

We express our sincere appreciation to the anonymous reviewers for their constructive feedback and invaluable suggestions. Their insightful comments have greatly contributed to the refinement and clarity of this paper.

References

Fabrice Colas and Pavel Brazdil. 2006. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Daniele Croce, Domenico Garlisi, and Marco Siino. 2022. An SVM ensemble approach to detect irony and stereotype spreaders on twitter. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2426–2432. CEUR-WS.org.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29,*

2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1746–1751. ACL.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024a. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2024b. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, 5(3):1339–1348.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Joshua Littenberg-Tobias, G. R. Marvez, Garron Hillaire, and Justin Reich. 2022. Comparing few-shot learning with GPT-3 to traditional machine learning approaches for classifying teacher simulation responses. In *AIED (2)*, volume 13356 of *Lecture Notes in Computer Science*, pages 471–474. Springer.

Francesco Lomonaco, Gregor Donabauer, and Marco Siino. 2022. COURAGE at checkthat!-2022: Harmful tweet detection using graph neural networks and ELECTRA. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 573–583. CEUR-WS.org.

Francesco Lomonaco, Marco Siino, and Maurizio Tesconi. 2023. Text enrichment with japanese language to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2708–2716. CEUR-WS.org.

Beatriz Matias Santana Maia, Maria Clara Falcão Ribeiro de Assis, Leandro Muniz de Lima, Matheus Becali Rocha, Humberto Giuri Calente, Maria Luiza Armini Correa, Danielle Resende Camisasca, and Renato Antonio Krohling. 2024. Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer. *Expert Systems with Applications*, 241:122418.

Stefano Mangione, Marco Siino, and Giovanni Garbo. 2022. Improving irony and stereotype spreaders detection using data augmentation and convolutional neural network. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2585–2593. CEUR-WS.org.

Zong Meng, Zhaohui Zhang, Yang Guan, Jimeng Li, Lixiao Cao, Meng Zhu, Jingjing Fan, and Fengjie Fan. 2024. A hierarchical transformer-based adaptive metric and joint-learning network for few-shot rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(3).

Mohsen Miri, Mohammad Bagher Dowlatshahi, Amin Hashemi, Marjan Kuchaki Rafsanjani, Brij B Gupta, and W Alhalabi. 2022. Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*, 37(12):11319–11341.

Fuad Muftie and Muhammad Haris. 2023. Indobert based data augmentation for indonesian text classification. In *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, page 128 – 132.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE access*, 7:100943–100953.

Marco Siino. 2024a. Badrock at semeval-2024 task 8: Distilbert to detect multigenerator, multidomain and multilingual black-box machine-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024b. Mcrock at semeval-2024 task 4: Mistral 7b for multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024c. T5-medical at semeval-2024 task 2: Using t5 medical embeddings for natural language inference on clinical trial data. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2021. Detection of hate speech spreaders using convolutional neural networks. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 2126–2136. CEUR-WS.org.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2022a. Fake news spreaders detection: Sometimes attention is not all you need. *Information*, 13(9):426.

Marco Siino, Marco La Cascia, and Ilenia Tinnirello. 2022b. Mcrock at semeval-2022 task 4: Patronizing and condescending language detection using multi-channel cnn, hybrid lstm, distilbert and xlnet. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 409–417. Association for Computational Linguistics.

Marco Siino, Francesco Lomonaco, and Paolo Rosso. 2024a. Backtranslate what you are saying and i will tell who you are. *Expert Systems*, n/a(n/a):e13568.

Marco Siino, Maurizio Tesconi, and Ilenia Tinnirello. 2023. Profiling cryptocurrency influencers with few-shot learning using data augmentation and ELECTRA. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2772–2781. CEUR-WS.org.

Marco Siino and Ilenia Tinnirello. 2023. Xlnet with data augmentation to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2763–2771. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2022. T100: A modern classic ensemble to profile irony and stereotype spreaders. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2666–2674. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024b. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

José Medardo Tapia-Téllez and Hugo Jair Escalante. 2020. Data augmentation with transformers for text classification. In *Advances in Computational Intelligence*, pages 247–259, Cham. Springer International Publishing.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Dang Van Thin, Duong Ngoc Hao, and Ngan Luu-Thuy Nguyen. 2023. Vietnamese sentiment analysis: An overview and comparative study of fine-tuning pretrained language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Trans. Circuits Syst. Video Technol.*, 33(12):7789–7802.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

0x.Yuan at SemEval-2024 Task 2: Agents Debating can reach consensus and produce better outcomes in Medical NLI task

Yu-An Lu
National Chupei High School
luyuam0@gmail.com

Hung-Yu Kao
National Cheng Kung University
hykao@mail.ncku.edu.tw

Abstract

In this paper, we introduce a multi-agent debating framework, experimenting on SemEval 2024 Task 2. This innovative system employs a collaborative approach involving expert agents from various medical fields to analyze Clinical Trial Reports (CTRs). Our methodology emphasizes nuanced and comprehensive analysis by leveraging the diverse expertise of agents like Biostatisticians and Medical Linguists. Results indicate that our collaborative model surpasses the performance of individual agents in terms of Macro F1-score. Additionally, our analysis suggests that while initial debates often mirror majority decisions, the debating process refines these outcomes, demonstrating the system's capability for in-depth analysis beyond simple majority rule. This research highlights the potential of AI collaboration in specialized domains, particularly in medical text interpretation.

1 Introduction

Clinical Trial Reports (CTRs) are indispensable in clinical research, providing critical data that reveal the efficacy of new treatments on patients. However, the exponential growth in the volume of CTRs, due to the increase in clinical trials, challenges researchers in conducting individual report analyses. With the swift progress in Natural Language Processing (NLP) technologies, leveraging machine learning algorithms for automating the review of CTRs is increasingly recognized as a feasible and promising solution (Saban et al., 2024)(Amar et al., 2024).

For SemEval-2024 Task 2 (Jullien et al., 2024a), the organizers introduced an English dataset derived from CTRs (Jullien et al., 2023), aimed at evaluating the truthfulness of CTR-statement pairs by discerning their veracity. This dataset includes a series of CTRs alongside associated statements, each designed to represent a hypothesis that must be classified as either Entailment or Contradiction, based on its alignment with the CTR content.

In addressing this intricate challenge, our study introduces a novel multi-agent debating framework. Characterized by a diverse assemblage of expert agents – including but not limited to a Bio-Statistician, Medical Linguist, and Pharmacologist – this system facilitates structured debates to adjudicate on the classification of each statement as an entailment or contradiction. By harnessing the distinctive expertise and viewpoints of various agents, we significantly augment the precision and dependability of our assessments. Our observations indicate that consensus among agents typically emerges within the second or third round of discussion, with agents exhibiting varied opinions on the statements under review. This multi-agent debate approach has demonstrably surpassed the outcomes achievable through single-agent or direct Large Language Model (LLM) interventions. Despite not achieving top-tier placement on the leaderboard, largely due to our adoption of a zero-shot approach without model fine-tuning, our system's broad applicability across different domains remains a compelling advantage.

2 Background

2.1 Related Works

Large Language Models (LLMs) LLM represent a significant stride in machine learning, offering the capability to generate coherent natural language text based on given contexts (Shanahan, 2023). The advent of InstructGPT (Ouyang et al., 2022) epitomizes this progression, heralding a new era of LLMs with enhanced instruction-following and logical reasoning skills. Although proprietary models like OpenAI's GPT-3.5 and GPT-4 set performance benchmarks, the rise of open-source LLMs presents a compelling narrative of achieving comparable state-of-the-art (SOTA)

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 305–310

performance with cost-effective implementations (Li et al., 2023)(Jiang et al., 2024).

Multi-Agent Collaboration Drawing parallels to human teamwork, integrating LLMs as collaborative agents has shown improved efficacy across diverse tasks. Initiatives like BabyAGI (Nakajima, 2023) introduced frameworks for automatic task generation and execution, based on predefined objectives. AutoGPT (aut, 2023) extends LLMs' capabilities to interact with external tools for executing real-world tasks, such as web scraping and code execution. Furthermore, HuggingGPT (Shen et al., 2023) functions as a model selector within the Hugging Face ecosystem, optimizing task-specific model selection. MetaGPT (Hong et al., 2023) emulates a software development team, assigning distinct roles to LLMs to streamline the design and development process. This body of work underscores the significant enhancements and novel functionalities afforded by multi-agent collaboration.

LLM Debating System Debates, a cornerstone in assessing the viability of ideas within human discourse, have been adapted to the realm of LLMs. Initial investigations by (Liang et al., 2023) into multi-agent debating revealed that a structured, mildly antagonistic debate could refine LLM outputs. Subsequent research (Xiong et al., 2023) corroborated the potential of LLMs to achieve consensus through debate. However, studies by (Chen et al., 2023) and (Agashe et al., 2023) on the evaluation of multi-agent debating systems highlighted a critical issue: the risk of consensus being swayed by majority opinion rather than individual agent analysis. This introduces an element of uncertainty regarding whether the consensus reached is genuinely reflective of a reasoned agreement or merely a product of majority rule. This paper seeks to explore and address this ambiguity in the context of LLM debating systems.

2.2 Dataset Discription

The dataset for our study, meticulously curated by clinical domain experts, trial organizers, and research oncologists affiliated with the Cancer Research UK Manchester Institute and the Digital Experimental Cancer Medicine Team (Jullien et al., 2023), comprises the following elements:

- **1–2 CTRs**: Record some key information during clinical trial, constitute by these four parts:
  - **Eligibility Criteria:** Specifies the required conditions for patients to participate in the clinical trial.
  - **Intervention Details:** Outlines the type, dosage, frequency, and duration of the treatments under study.
  - **Trial Results:** Details the number of participants, outcome measures, measurement units, and the observed results.
  - **Adverse Events Reporting:** Records any symptoms or signs noted in patients during the course of the clinical trial.
- **Statement**: An assumption based on CTRs, which hasn't been verified to be correct or not
- **Section Marker**: Which section in the CTRs is the statement based on.
- **Entailment/Contradiction label**: The statement is Entailment/Contradiction to the CTRs.

Table 1 describe the constitute of the dataset, and Table 2 is a example of test data.

Dataset	Comparison	Single	Total
Train	665	1035	1700
Dev	60	140	200
Test	2947	2553	5500

Table 1: Constitute of the dataset.

Attribute	Value
Type	Single
Section_id	Results
Primary_id	NCT02640053
CTR_context	Outcome Measurement: Area Under the Curve (AUC) EORTC CIPN20 Sensory Neuropathy Subscale...(omitted)
Statement	Patients in the primary trial that didn't receive topical cryotherapy had worse symptoms than patients that did receive topical cryotherapy.
Label	Contradiction

Table 2: Test data example.

3 System Overview

3.1 Motivation

Existing multi-agent collaboration frameworks, while adept at executing tasks like coding, often fall short in fostering substantive dialogues among agents. This limitation hinders the development of critical thinking skills, as agents are not encouraged to engage in detailed discussions or critically evaluate one another's viewpoints. Recognizing this gap, we introduce a novel framework designed specifically to enable multi-agent debate. Our approach centers on facilitating a collaborative environment where agents are encouraged to thoroughly consider and reflect on the perspectives of their counterparts. By prioritizing in-depth discussions and critical analysis, we aim to advance the capabilities of multi-agent systems beyond mere task execution to include nuanced, critical deliberations.

3.2 Multi-Agent Debating Framework

The multi-agent debating framework constitute by several costume *agents*, a *issue* to determine and a *logical judgment unit*. Below Algorithm 1 are pseudocode that describe how the framework operates:

Algorithm 1: LLM Multi-Agent Debate Framework

Data: *Issue* to be debated, *Agents* involved in the debate
Result: Conclusive outcome(**Entailment** or **Contradiction**)

initialization: Set turn $t_i = 0$;
Agents generate initial responses r_i with ***Opinion*** and ***Decision***;
while *not reached maximum number of turns and no consensus* **do**
    Assess consensus among ***Opinions***;
    if *consensus* **then**
        Adopt ***Opinions*** as outcome and terminate;
    else
        Increment turn t_{i+1};
        Update agents with others' ***Decisions***;
        Agents revise responses r_{i+1};
    end
end
if *no consensus after maximum turns* **then**
    Take most ***Opinions*** as final result;
end

Upon presenting an issue, the framework, in its initial turn denoted as t_i, solicits from agents the generation of an initial response r_i. Each response is required to concurrently encompass ***Opinion*** and ***Decision***, wherein ***Opinion*** constitutes a paragraph articulating the agent's stance on the issue, and ***Decision*** represents one of two potential outcomes: Entailment or Contradiction. Subsequent to the formulation of responses by all agents, the logical judgment unit assesses the presence of consensus within their opinions. In the event of consensus, their ***Opinions*** are adopted as the conclusive outcome, thereby terminating the framework. Conversely, in the absence of consensus, the debate advances to the subsequent turn t_{i+1}. In this phase, each agent is apprised of the ***Decisions*** made previously by other agents and is prompted to generate a revised response r_i. This iterative process persists until a consensus is established among the agents, or upon reaching the predefined maximum number of turns, at which point the framework is concluded and the amalgamation of multiple ***Opinions*** is deemed the final result.

4 Experiments

Our experiments were conducted using the Mixtral-8x7B model (Jiang et al., 2024), selected for its exceptional performance and cost-efficiency.

4.1 Sections Select

The task of pinpointing the relevant sections within Clinical Trial Reports (CTRs) for statement verification was entrusted to a Large Language Model (LLM). This procedure entailed providing the LLM with a detailed prompt, encompassing explicit instructions, the statement under scrutiny, and the entirety of the CTR text. The LLM's assignment was to ascertain the sections of the CTR pertinent to the statement. An example of the LLM's output is delineated below, illustrating its capability to effectively identify and isolate relevant text segments.

Listing 1: LLM's output to select sections

```
{
  "Primary_CT": {
    "Adverse Events": true,
    "Results": false,
    "Eligibility": true,
    "Intervention": false
  }
}
```

4.2 Agents Design

We designed five agents, which are all experts in medical field:

- **Dr. Emily Nguyen**: Biostatistician focusing on data interpretation and analysis in clinical trials.
- **Dr. Alex Johnson**: Medical Linguist specializing in clinical text analysis and medical jargon clarification.
- **Dr. Aisha Patel**: Pharmacologist dedicated to drug action understanding and safety evaluation in trials.
- **Dr. Liang Wei**: Epidemiologist studying health and disease patterns in populations for disease control.
- **Dr. Maria Gomez**: Cardiologist treating cardiovascular diseases and managing heart-related conditions.

5 Results

5.1 Official Evaluation Metrics

SemEval-2024 Task 2 organizers had mentioned several evaluation metrics: Macro F1-score, Faithfulness and Consistency(Jullien et al., 2024a), we will use these metrics to evaluate the result.

- **Faithfulness**: Quantifies the precision with which a system arrives at the correct conclusion based on the right reasons. Assessed by examining the system's ability to adjust its predictions accurately in response to semantic alterations. Evaluated using N statements x_i from a contrast set (C), their related original statements y_i, and the model's predictions $f()$.
- **Consistency**: Measures a system's capacity to produce identical outputs for semantically equivalent scenarios. Determined by the system's capability to consistently predict the same label for both original and contrast statements, even in the case of semantically preserving interventions. This involves analyzing N statements x_i in a contrast set (C), their corresponding original statements y_i, and the model's predictions $f()$.

Agent	F1	Fth	Con
Debated	**0.69**	0.51	0.56
Dr. Emily Nguyen	0.65	**0.55**	**0.58**
Dr. Alex Johnson	0.67	0.52	0.56
Dr. Aisha Patel	0.67	**0.55**	0.57
Dr. Liang Wei	0.66	0.53	0.57
Dr. Maria Gomez	0.66	0.53	0.57

F1 = Macro-F1, Fth = Faithfulness, Con = Consistency

Table 3: Performance metrics of different agents.

5.2 Impact of Multi-Agent Debates on Performance

Table 3 delineates the comparative performance metrics on the test dataset for decisions made by individual agents versus the consolidated outcome of the multi-agent debating, as evaluated on the official Codalab platform (Jullien et al., 2024b). The data distinctly showcases that the integration of multi-agent debating significantly enhances the F1-score, surpassing the performance of any single agent's decision-making capabilities.

Regarding our standing in the competition, our team, identified as "Yuan_Lu," achieved the 22nd position out of 32 participating teams.

5.3 Demonstration of Debating Process

The excerpt below showcases the dynamic of our multi-agent debating process, wherein agents engage in a layered analysis of the problem from their unique vantage points. Notably, our methodology facilitates not just the articulation of each agent's perspective but also the receptiveness to alternate viewpoints. This iterative exchange allows agents to both influence and be influenced, showcasing the framework's capacity for fostering a comprehensive and nuanced understanding through collaborative debate.

5.3.1 Turn 1

DR. EMILY NGUYEN: Based on the data provided, I cannot confirm the statement about the occurrence of asthenia and pyrexia... **The data does not include any information on these adverse events** for either the primary or secondary clinical trials... So, my decision is: **Contradiction**

DR. ALEX JOHNSON: After carefully reviewing the data provided and considering the opinions expressed, I agree that **there is**

no evidence in the data to support the statement about the occurrence of asthenia and pyrexia... So, my decision is: **Contradiction**

5.3.2 Turn 2

DR. EMILY NGUYEN: After further discussion and considering the perspectives shared, **I now see the importance of rigorous data analysis and transparent reporting**. This ensures healthcare professionals and patients have a complete understanding... In summary, **the provided data does not support the statement**... So, my decision is: **Contradiction**

DR. ALEX JOHNSON: Reflecting on the discussions, I acknowledge my initial decision might have been premature. **Given the lack of evidence in the provided data**, I adjust my stance... Therefore, **I align with the consensus that the statement is a contradiction**... So, my decision is: **Contradiction**

5.4 Beyond Majority Rule: The Dynamics of Multi-Agent Debating

To examine the effectiveness of the multi-agent debating framework, we introduce two key metrics: the F1 score of the majority decision in the first round of debate, and the rate of alignment between the initial majority decision and the final decision.

- **F1 Score of Correct Majority Decision**: This metric measures the precision and recall of the majority decision in the first round of debate. The calculated F1 score of 0.567 suggests moderate accuracy of the initial majority opinion. This indicates that while the majority decision is somewhat aligned with the correct labels, there is still considerable room for improvement.

- **Rate of Alignment Between First Majority and Final Decision**: This metric reflects the consistency between the initial majority decision and the final decision after the debate. A high rate of 91.93% suggests that the final decisions of the multi-agent system are predominantly in line with the initial majority opinions. This high alignment rate might indicate a strong influence of the initial majority view on the final outcome of the debate.

Considering these results, the multi-agent debating framework demonstrates a significant tendency to converge towards the initial majority decision. However, the fact that the final decision's F1 score is 0.69, which is higher than the initial majority's F1 score, indicates that the debating process adds value beyond simply following the majority rule. This suggests that while the final decision often aligns with the initial majority opinion, the debate process itself contributes to refining the decision, potentially correcting or enhancing the initial judgment. Therefore, despite the high alignment rate, the multi-agent debating framework plays a critical role in facilitating a more comprehensive and informed decision-making process.

6 Conclusion

We have presented a novel multi-agent debating framework to participate SemEval-2024 Task 2. This approach, integrating the expertise of diverse agents like Biostatisticians, Medical Linguists, and Pharmacologists, significantly enhances the analysis of Clinical Trial Reports (CTRs). Our findings demonstrate improved performance in entailment or contradiction determination of CTR-statement pairs, as evidenced by enhanced Macro F1-scores compared to individual agent assessments. Despite a tendency to align with initial majority decisions, the debating process refines these initial judgments, indicating the framework's effectiveness beyond simple majority rule.

References

2023. Auto-gpt.documentation.

Saaket Agashe, Yue Fan, and Xin Eric Wang. 2023. Evaluating multi-agent coordination abilities in large language models.

F. Amar, A. April, and A. Abran. 2024. Electronic health record and semantic issues using fast healthcare interoperability resources: Systematic mapping review. *Journal of Medical Internet Research*, 26:e45209.

Huaben Chen, Wenkang Ji, Lufeng Xu, and Shiyu Zhao. 2023. Multi-agent consensus seeking via large language models.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2023. Metagpt: Meta programming for a multi-agent collaborative framework.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Mael Jullien, Marco Valentino, and André Freitas. 2024a. Semeval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1948–1963, Mexico City, Mexico. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, and André Freitas. 2024b. CodaLab - Competition — codalab.lisn.upsaclay.fr. `https://codalab.lisn.upsaclay.fr/competitions/16190`. [Accessed 13-02-2024].

Shiyang Li, Jun Yan, Hai Wang, Zheng Tang, Xiang Ren, Vijay Srinivasan, and Hongxia Jin. 2023. Instruction-following evaluation through verbalizer manipulation.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate.

Yohei Nakajima. 2023. Babyagi. GitHub repository.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

M. Saban, M. Lutski, I. Zucker, M. Uziel, D. Ben-Moshe, A. Israel, S. Vinker, A. Golan-Cohen, I. Laufer, I. Green, R. Eldor, and E. Merzon. 2024. Identifying diabetes related-complications in a real-world free-text electronic medical records in hebrew using natural language processing techniques. *Journal of diabetes science and technology*, page 19322968241228555.

Murray Shanahan. 2023. Talking about large language models.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate.

TW-NLP at SemEval-2024 Task10: Emotion Recognition and Emotion Reversal Inference in Multi-Party Dialogues.

Wei Tian[1], **Peiyu Ji**[2], **Yuan Zheng**[1], **Lei Zhang**[1], **Yue Jian**[1]
[1]Beijing Smartdot Technology Co., Ltd, China
[2]Zhongyuan University of Technology, China
tianwei@smartdot.com,
2020107223@zut.edu.cn

Abstract

In multidimensional dialogues, emotions serve not only as crucial mediators of emotional exchanges but also carry rich information. Therefore, accurately identifying the emotions of interlocutors and understanding the triggering factors of emotional changes are paramount. This study focuses on the tasks of multilingual dialogue emotion recognition and emotion reversal reasoning based on provocateurs, aiming to enhance the accuracy and depth of emotional understanding in dialogues. To achieve this goal, we propose a novel model, MBERT-TextRCNN-PL, designed to effectively capture emotional information of interlocutors. Additionally, we introduce XGBoost-EC (Emotion Capturer) to identify emotion provocateurs, thereby delving deeper into the causal relationships behind emotional changes. By comparing with state-of-the-art models, our approach demonstrates significant improvements in recognizing dialogue emotions and provocateurs, offering new insights and methodologies for multilingual dialogue emotion understanding and emotion reversal research.

1 Introduction

The EDiReF shared task at SemEval 2024 encompasses three subtasks(Kumar et al., 2024): Task 1 involves Emotion Recognition (ERC) in mixed Hindi-English dialogues, Task 2 focuses on Emotion Flipping Reasoning (EFR) in mixed Hindi-English dialogues, and Task 3 involves EFR in English dialogues. In Task 1 ERC, the goal is to assign emotions to each utterance in the dialogue, while in Task 2 and Task 3 EFR, the aim is to identify trigger utterances leading to emotion flipping in multi-party dialogues. The definitions of these tasks provide a crucial framework for understanding the dynamics of emotions in natural language conversations.

Firstly, we are committed to addressing two subtasks: Emotion Recognition in Conversations (ERC) in Task 1, and Emotion Flipping Reasoning (EFR) in Tasks 2 and 3. For Task 1, we constructed the MBERT-TextRCNN-PL model based on MBERT(Pires et al., 2019) and Prompt Learning to identify emotions in mixed-language dialogues. By leveraging the multilingual capability of the MBERT model and incorporating Prompt Learning, we successfully guided the model to focus on key aspects of emotion recognition. This approach can effectively handle mixed Hindi and English conversations while achieving the sharing of model parameters between different languages. Finally, to improve the robustness of the model, this paper integrates FGM to enhance the model's generalization ability.

Secondly, for Tasks 2 and 3, we proposed the XGBoost-EC (Emotion Capture) method aimed at identifying triggers of emotion flipping. We segmented dialogues into fixed-size windows and extracted emotion encodings from each window. To better encode the emotions of the final speaker, we used -1 to fill in blank emotion states within the window. Then, we employed the XGBoost algorithm(Chen and Guestrin, 2016) for classification to identify windows that could potentially be triggers of emotion flipping. By guiding the model to learn patterns and features of emotion flipping triggers through annotated data during training, we were able to effectively identify triggers of emotion flipping in dialogues.

In the EDiReF shared task at SemEval 2024, our[1] proposed MBERT-TextRCNN-PL model achieved 6th place in Task 1. Additionally, our XGBoost-EC model secured 1st place in Task 2 and 5th place in Task 3.

[1]Our codes are available at https://github.com/TW-NLP/SemEval2024-Task10

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 311–315

2 Background

2.1 Dataset Description

Task 1, the Emotion Recognition (ERC) task, corresponds to the MASAC-ERC dataset compiled by extracting dialogues from Indian television dramas. The dataset comprises 446 dialogues, with 8 emotion categories: disgust, contempt, anger, neutral, joy, sadness, fear, and surprise. The training set consists of 343 dialogues containing 8,506 sentences, the validation set consists of 45 dialogues containing 1,354 sentences, and the test set consists of 56 dialogues containing 1,580 sentences, The data analysis for Task1 is shown in Table 1.

Set	Dlgs	Utts
Train	343	8506
Val	45	1354
Test	56	1580

Table 1: Data statistics for Task1.

For Task 2, corresponding to MASAC-EFR, the dataset includes 5,667 dialogues. The training set comprises 4,893 dialogues containing 98,777 sentences, the validation set comprises 389 dialogues containing 7,462 sentences, and the test set comprises 385 dialogues containing 7,690 sentences, The data analysis for Task2 is shown in Table 2.

Set	Dlgs	Utts
Train	4893	98777
Val	389	7462
Test	385	7690

Table 2: Data statistics for Task2.

Regarding Task 3, corresponding to MELD-EFR, the dataset consists of 5,428 dialogues. The training set comprises 4,000 dialogues containing 35,000 sentences, the validation set comprises 426 dialogues containing 3,522 sentences, and the test set comprises 1,002 dialogues containing 8,642 sentences, The data analysis for Task3 is shown in Table 3.

Set	Dlgs	Utts
Train	4000	35000
Val	426	3522
Test	1002	8642

Table 3: Data statistics for Task3.

2.2 Related Work

Emotion Recognition in Conversation. Emotion recognition in dialogues is categorized into monolingual and multilingual dialogue emotion recognition. Under monolingual conditions, the DialogXL(Shen et al., 2021) model utilizes the XL-Net(Yang et al., 2019) architecture for Emotion Recognition in Conversation (ERC). They encode dialogue discourse and leverage dialogue-aware self-attention to incorporate dialogue semantics. Additionally, (Jiao et al., 2019) employs a hierarchical gated recursive unit framework involving two different levels of GRU. The lower-level GRU models word-level inputs, while the higher-level GRU captures contextual information at the discourse level. Furthermore, (Lian et al., 2021) proposes a correction model named "Dialogue Emotion Correction Network (DECN)." The aim of this work is to enhance emotion recognition performance by automatically identifying errors made by emotion recognition strategies. (Shou et al., 2022) employs graph-based approaches to tackle ERC. They introduce a session-level sentiment analysis model that combines dependency parsing and graph convolutional neural networks. Self-attention mechanisms capture the most effective words in the dialogue and then construct a graph. In multilingual settings, (Kumar et al., 2023a) proposes an advanced fusion technique that first translates into a uniform language, followed by the integration of commonsense knowledge with the dialogue comprehension module.

Emotion Flipping Reasoning. (Kumar et al., 2022)introduces a novel Emotional Flip Reasoning (EFR) aimed at identifying the utterance that triggered an emotional state flip in an individual at a certain point in the past. Additionally, a Transformer-based network is proposed to carry out the Emotional Flip Reasoning. To identify emotional instigators, (Kumar et al., 2023b) proposes the TGIF framework for multilingual dialogue data. It utilizes a combination of Transformer and GRU to identify emotional instigators.

3 System Overview

3.1 ERC

In Task 1's data format, we designed the MBERT-TextRCNN-PL model based on prompt learning, as illustrated in Figure 1, to contextualize the conversation. The input data for this model is organized according to the format in Table 4.

Speaker	Utterance	Emotion
Sp1	Aaj to bhot awful day tha!	Sad
Sp2	Oh no! Kya hua?	Sad
Sp1	Kisi ne mera sandwich kha liya!	Sad
Sp2	Me abhi tumhare liye new bana deti hun!	Joy

Table 4: Data instance for ERC task

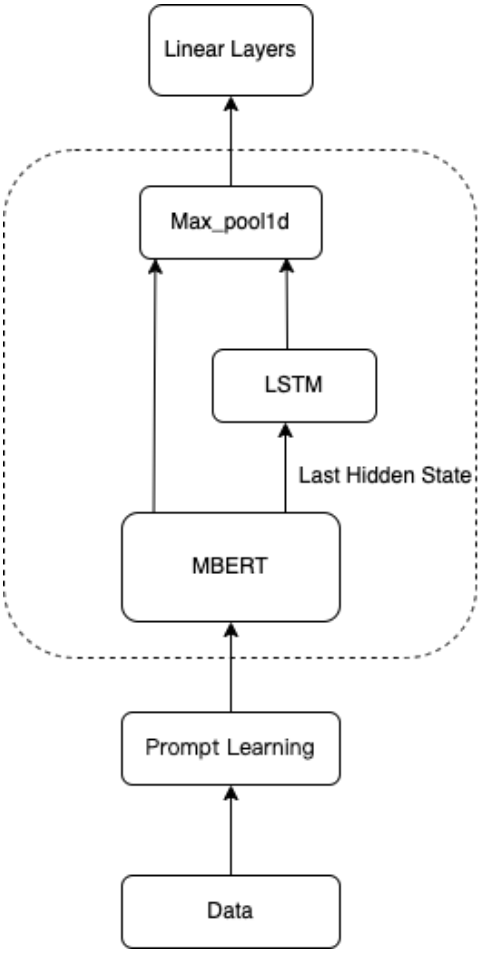


Figure 1: The architecture diagram of MBERT-TextRCNN-PL.

Specifically, we employ prompt learning to construct the model's input, where the inputs for Sp1 and Sp2 are formatted as follows: "The following is 's conversation history. Sp1: Aaj to bhot awful day tha!" and "The following is 's conversation history. Sp1: Aaj to bhot awful day tha!, Sp2: Oh no! Kya hua?". Through this formatted input, the model can better understand the conversation history and context, thereby comprehensively capturing semantic correlations in the text.

Building upon this, we propose a vector representation approach that combines the features of MBERT and TextRCNN. Firstly, we utilize the MBERT pre-trained model to encode the text sequences into semantic vectors, leveraging its multilingual semantic understanding capability. Subsequently, contextual information of the sequences is further enhanced by capturing it through a bidirectional LSTM layer, strengthening the model's comprehension of the dialogue history. Next, the last hidden state output of MBERT and the output of LSTM are concatenated in the feature dimension to fuse word-level and sequence-level semantic information. Finally, classification of the text data is achieved through a fully connected layer, enabling effective categorization of the text.

This architecture fully leverages the multilingual semantic understanding of MBERT and the sequence modeling capability of TextRCNN, enabling the model to better understand and classify text data. Such design not only enhances the performance and effectiveness of the model in text understanding tasks but also improves its understanding of context, allowing the model to more accurately capture semantic information in the text.

3.2 EFR

For EFR's Task 2 and Task 3, we propose the XGBoost-EC model, as illustrated in Figure 2. The XGBoost-EC model is an emotion recognition model based on XGBoost, designed to utilize emotion feature encoding for training and prediction.

In the XGBoost-EC model, we initially encode the emotion features, converting them into numerical forms for processing by the XGBoost algorithm. Specifically, we map emotion labels to integer values, such as encoding 'joy' as 1, 'sadness' as 2, and so forth. To better capture emotion information, we employ emotion windows for encoding and assign -1 for missing emotions.

By encoding emotions, the XGBoost-EC model can learn the associations between emotions and other features, thereby predicting the emotional states in text or dialogue more accurately. Leveraging the efficient performance of the XGBoost algorithm and its capability to handle large-scale

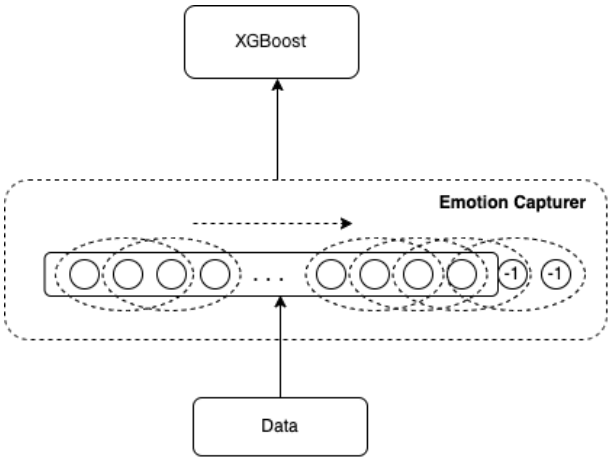


Figure 2: The architecture diagram of XGBoost-EC.

datasets, this model holds significant value for emotion recognition tasks.

4 Experimental Setup

Task 1. For the ERC Task 1, our model was built using the Hugging Face Transformers library, where we directly employed pre-trained tokenizer and language models for further fine-tuning. Specifically, we utilized the MBERT-TextRCNN-PL model with parameter configurations of a learning rate of 3e-6 and a batch size of 16 for training. We opted for the AdamW optimizer to update the model parameters. To ensure the model could handle longer text sequences, we set the maximum sequence length of the tokenizer to 512. During the model's validation evaluation process, we employed the F1 score as the performance metric. Hardware-wise, we utilized the NVIDIA RTX3090 (24G) graphics card to accelerate both model training and inference processes.

Task 2. For the EFR Task 2 and Task 3, we built the XGBoost-EC model based on the XGBoost library. In Task 2, after validation, we chose to set the scale-pos-weight parameter in the XGBClassifier to 1.08 to address the issue of imbalanced samples. Additionally, we fixed the random seed to 42 to ensure reproducibility of the results. To better capture changes in emotion, we set the emotion window size to 4.

Task 3. In Task 3, we adjusted the scale-pos-weight parameter in the XGBClassifier to 1.6 to accommodate different levels of sample imbalance. Similarly, we fixed the random seed to 42 and set the emotion window size to 3, enabling the model to capture trends in emotion changes more effectively.

5 Results

5.1 Task1

Based on Table 5, we observe that the performance of MBERT-TextRCNN-PL is more competitive compared to BERT, RoBERTa, MBert, MURIL, CoMPM, DialogXL, BERT+COFFEE and MBert+COFFEE as contrasted by Shivani Kumar et al(Kumar et al., 2023a). MBERT-TextRCNN-PL, based on prompt learning, provides a better summary of the former's remarks, which aligns more closely with conventional expression norms. Leveraging the multilingual MBERT-TextRCNN model enhances the representation of semantic information, rendering the model state-of-the-art. As a result, it achieved the 6th position on the official leaderboard.

Model	F1
BERT	0.40
RoBERTa	0.41
MBert	0.30
MURIL	0.35
CoMPM	0.35
DialogXL	0.41
BERT+COFFEE(Kumar et al., 2023a)	0.41
MBERT+COFFEE(Kumar et al., 2023a)	0.31
Ours(MBERT-TextRCNN-PL)	0.46

Table 5: The results of Task 1.

5.2 Task2

As shown in Table 6, the proposed XGB-EC achieved an F1 score of 0.79 in the evaluation of Task 2, securing the first position in this task.

Model	F1
TECHSSN Team	0.1
IASBS Team	0.12
IITK Team	0.56
Knowdee Team	0.66
FeedForward Team	0.77
Ours(XGB-EC)	0.79

Table 6: The results of Task 2.

5.3 Task3

As indicated in Table 7, the proposed XGB-EC outperformed AGHMN, TL-ERC, DGCN, DialogXL, and BERT by a significant margin, exhibiting a substantial improvement compared to the TGIF

framework with a 38-point increase in F1 score. In Task 3, it obtained the 5th position. The XGB-EC framework not only considers the current emotion but also captures emotion variations through emotion windows, enabling a better understanding of emotional provocations.

Model	P	R	F1
AGHMN	0.15	0.17	0.16
TL-ERC	0.07	0.33	0.13
DGCN	0.10	0.67	0.17
DialogXL	0.09	0.34	0.15
BERT	0.14	0.55	0.21
TGIF(Kumar et al., 2023b)	0.26	0.55	0.33
Ours(XGB-EC)	0.71	0.71	0.71

Table 7: The results of Task 3.

6 Conclusion

In this paper, we focus on addressing the challenges of multilingual conversation emotion recognition and emotion flipping reasoning tasks. To this end, we propose two models to tackle these tasks.

Firstly, for the multilingual conversation emotion recognition task, we introduce the MBERT-TextRCNN-PL model. This model combines prompt learning and the MBERT-TextRCNN approach to better recognize emotions from multiple speakers. Through prompt learning, we can provide richer contextual information, thereby accurately capturing the emotional content in the text. Leveraging the features of MBERT-TextRCNN, this model effectively utilizes the capabilities of multilingual semantic understanding and sequence modeling to enhance the accuracy and effectiveness of emotion recognition.

Secondly, for Task 2 and Task 3 of EFR, we propose the XGBoost-EC (emotion capturer) model, aimed at identifying emotion instigators. This model, employing emotion windows and XGBoost classifier, captures emotion instigators more effectively. The setting of emotion windows allows the model to consider the historical trend of emotion changes, leading to a more comprehensive analysis of emotional data. The application of the XGBoost classifier enables efficient classification and reasoning of emotional data, thereby enhancing the model's ability to identify instigators.

The introduction of these two models provides effective solutions for multilingual conversation emotion recognition and emotion flipping reasoning tasks, offering new insights and methods for research and applications in related fields.

References

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R Lyu. 2019. Higru: Hierarchical gated recurrent units for utterance-level emotion recognition. *arXiv preprint arXiv:1904.04446*.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024–task 10: Emotion discovery and reasoning its flip in conversation (ediref). *arXiv preprint arXiv:2402.18944*.

Shivani Kumar, Md Shad Akhtar, Tanmoy Chakraborty, et al. 2023a. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. *arXiv preprint arXiv:2310.13080*.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023b. Emotion flip reasoning in multiparty conversations. *arXiv preprint arXiv:2306.13959*.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Zheng Lian, Bin Liu, and Jianhua Tao. 2021. Decn: Dialogical emotion correction network for conversational emotion recognition. *Neurocomputing*, 454:483–495.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Weizhou Shen, Junqing Chen, Xiaojun Quan, and Zhixian Xie. 2021. Dialogxl: All-in-one xlnet for multiparty conversation emotion recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13789–13797.

Yuntao Shou, Tao Meng, Wei Ai, Sihan Yang, and Keqin Li. 2022. Conversational emotion recognition studies based on graph convolutional neural networks and a dependent syntactic analysis. *Neurocomputing*, 501:629–639.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

UWBA at SemEval-2024 Task 3: Dialogue Representation and Multimodal Fusion for Emotion Cause Analysis

Josef Baloun **Jiří Martínek** **Ladislav Lenc**
New Technologies for The Information Society, University of West Bohemia, Pilsen
{balounj,jimar,llenc}@ntis.zcu.cz

Pavel Král **Matěj Zeman** **Lukáš Vlček**
Department of Computer Science and Engineering, University of West Bohemia, Pilsen
{pkral,zemanm98,vlcek0}@kiv.zcu.cz

Abstract

In this paper, we present an approach for solving SemEval-2024 Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations. The task includes two subtasks that focus on emotion-cause pair extraction using text, video, and audio modalities. Our approach is composed of encoding all modalities (MFCC and Wav2Vec for audio, 3D-CNN for video, and transformer-based models for text) and combining them in an utterance-level fusion module. The model is then optimized for link and emotion prediction simultaneously. Our approach achieved 6th place in both subtasks. The full leaderboard can be found at https://codalab.lisn.upsaclay.fr/competitions/16141#results.

1 Introduction

The *SemEval-2024 Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations* (Wang et al., 2024) is aimed at extracting emotion-cause pairs (ECPs) in conversations. The main data source to tackle this task is recordings from the sitcom *Friends* in the English language – Emotion-Cause-in-Friends dataset Wang et al. (2022).

The detection of emotions and a deeper analysis of what causes them is one of the interesting and important tasks that have recently been tackled within the NLP community. Previously, researchers focused their efforts on text-only emotion-cause extraction (Gui et al., 2018; Gao et al., 2017; Bostan et al., 2019). However, representing dialogues solely through the text (speech transcription) is not entirely adequate, as people use different intonations and other prosodic features. Moreover, the fact that something happens during the conversation (e.g. someone walks in or something breaks) affects the dialogue in terms of emotions and their causes. We can also mention different facial expressions for different emotions. So, the fact that a conversation in its natural form is multimodal (text, audio and video) opens a big space for research.

Our approach is targeted at both subtasks of the above-mentioned Semeval 2023 task. The main difference between them is the number of different modalities used for predicting ECPs. For Subtask 1, the prediction of ECPs is solely based on the text transcription without recordings. The goal is to provide text spans along with ECPs: i.e. the segment of an utterance primarily responsible for emotion-cause. Subtask 2, does not require extracted text spans (in some cases it is impossible because the emotion-cause is not expressed in the textual form), but it is required to use other modalities embedded in available `mp4` video files to extract emotion-cause pairs together with a target emotion.

Our approach uses all modalities and encodes them into a common utterance-level representation, which is then used for *link* and *emotion* prediction. The objective is to learn both prediction tasks with two loss functions that are combined. The main idea behind this is that emotions might positively influence the links (pairs) and vice versa.

Another point we would like to highlight is that the textual input is a whole dialogue, so the context is taken into consideration. This improves the link results significantly, as shown further. After validating our codes by the SemeEval task organizers, the final implementation will be released[1].

2 Task and Background

The goal of the emotion-cause pair extraction (ECPE) task (Xia and Ding, 2019) is to extract potential pairs of emotions and corresponding causes in a conversation/document and/or other source of dialogue. It is an extension of the emotion-cause extraction – ECE task (Lee et al., 2010), where the goal is to decide if a clause/utterance is the corresponding cause, given the annotation of emotions.

[1]https://github.com/martinekj/semeval_2024_Task3_ECPE

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 316–325

This SemEval task does not require only the extraction of corresponding pairs but also the prediction of emotions. In other words, extracted pairs must be complemented by the prediction of the target utterance emotion. So, for the evaluation, we have a triplet (a source utterance, a target utterance, an emotion of the target utterance). E.g., **3_joy, 2** which means that the emotion *joy* in *utterance 3* is caused by *utterance 2*.

Our team participated in both competition subtasks. We aimed to extract not only the pairs (as illustrated in the previous example) but also text spans – the exact parts of utterances/clauses primarily responsible for the emotion-cause (e.g. **3_joy, 2_You made up!** – meaning that the emotion *joy* in *utterance 3* is caused by the text span *You made up!* in *utterance 2*)

Hence, we work with all input data (i.e., multimodal – text, sound, image sequence/video) of the dataset **Emotion-Cause-in-Friends** that serves as the competition dataset.

2.1 Related Work

Lee et al. (2010) presented a text-based approach for the ECPE task. They created a rule-based system and tested it on a Chinese dataset created from the Sinica corpus.

Chen et al. (2020) proposed an approach that takes the ECPE task as a unified sequence labeling task. Their method combines a convolutional network with two bi-directional long short-term memory networks. They show that the approach outperforms several baselines. However, the score is slightly lower than baselines including BERT.

Poria et al. (2018) created the multimodal MELD dataset as an extension of the EmotionLines corpus and performed a baseline evaluation of the emotion recognition task on this data. Another multimodal dataset is presented in (Firdaus et al., 2020).

Wang et al. (2022) presented a multimodal approach for the emotion-cause pair extraction. The authors created a dataset including text, audio and video modalities. The baseline approach obtained F1 score of 0.51.

3 System Overview

We decomposed the main objective into emotion and link prediction (the estimation of pairs) tasks. The final result then consists of source and target utterances provided by the link and emotion of the target utterance.

The architecture is depicted in Figure 1. First, we encode the different modalities at the utterance or dialogue level to incorporate more context. Next, we fuse the representations at the utterance level. Once we have representations of all individual utterances in a dialogue, we predict links and emotions (Subtask 2). Based on this output, we employ our separate model for text span prediction, which is necessary for Subtask 1.

We have followed the competition rules and used pre-trained language models (PLMs), but no additional training data.

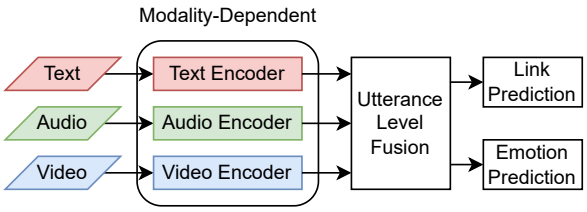


Figure 1: System architecture

3.1 Text Encoder

We employed a transformer-based encoder, such as BERT (Devlin et al., 2019), for text encoding. As depicted in Figure 2, the input consists of an entire dialogue. It commences with a CLS token and proceeds with tokens representing individual utterances. As usual, positional encoding corresponding to token position is applied. The utterances are separated by a SEP token, and the input is further extended by utterance embeddings. After encoding the tokens, we average the tokens of every single utterance to derive its representation. This way, the dialogue context is available in every single utterance.

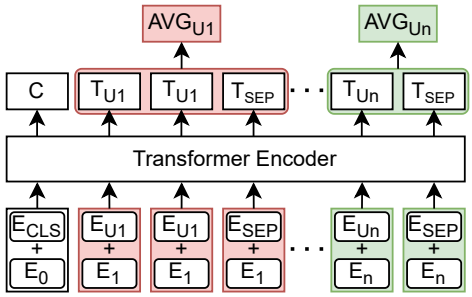


Figure 2: Text Encoder

3.2 Audio Encoder

For encoding the audio, we evaluated two methods. The first is referred to as *MFCC* feature extraction

and is based on Mel frequency cepstral coefficients (Tiwari, 2010). The second method is based on *Wav2Vec* model (Baevski et al., 2020).

3.2.1 MFCC Feature Extraction

We firstly denoise the audio files removing the background noise (background laughing track, people speaking in the café, etc.). We use the REPET-SIM (Rafii and Pardo, 2013) separation method to separate the main speaker voice line. The separated main speaker voice line audio is then used for computing the MFCC, and this audio representation is used in a long short-term memory (LSTM) model trained for the emotion recognition task. Audio feature vectors with a dimension of 2048 are acquired from the last hidden state of the model. The model comprises one bidirectional LSTM and two linear layers. The whole model is trained on the emotion recognition task.

The REPET-SIM method is a generalization of the REPET method (Rafii and Pardo, 2012). The basic idea behind the REPET method is to find repeating elements in audio, compare them with repeating models derived from them, and separate the repeating patterns through time-frequency masking. REPET-SIM specifically identifies these repeating elements by using a similarity matrix (Rafii and Pardo, 2013).

MFCC capture the shape of the power spectrum of a sound signal. They are computed by first transforming the audio into the frequency domain using the Discrete Fourier Transform and then applying the "mel" scale to approximate the human auditory perception of the sound frequency (Tiwari, 2010).

3.2.2 Wav2Vec

As an alternative, we used pre-trained version (Field, 2022) of Wav2Vec model as an audio encoder. It was fine-tuned for the emotion classification task, and subsequently, we conducted further fine-tuning on competition data. We averaged the audio sequence representations provided by the final layer resulting in a 1024-dimensional audio representation of the utterance. During the fine-tuning phase, the representation was utilized by a two-layer perceptron to predict emotion.

3.3 Video Encoder

We utilized the ResNext 3D-CNN (Hara et al., 2018) model with depth 101 pre-trained on the Kinetics (Carreira and Zisserman, 2017) dataset. For every 16 frames, this model provides an output vector with a dimension of 2048. In the case of longer videos, the final feature vector is computed with a global average pooling over the temporal dimension.

The input to the model are preprocessed image frames from the video file (using the *ffmpeg* python library). The preprocessing consists of scaling to 240x240 pixels (while preserving the aspect ratio with zero padding).

3.4 Fusion Module

The multimodal fusion relies on a transformer-based encoder. Since the dimensions of text, audio, and video representations may vary, they undergo linear projection using a linear layer with the fusion size (*fs*) as a parameter. Subsequently, they serve as tokens for the encoder input. The encoder consists of 6 layers with $fs/64$ heads and GELU activation function. The intermediate size is $4 \cdot fs$.

The fusion is done on the utterance level, so no explicit dialogue context is available, but it may be provided by encoded representations of individual modalities. For the fusion, we ignore the positional encoding.

We consider several fusion strategies. The first straightforward scenario is to use the aggregation function for each component of the encoded tokens: *AvgFusion* (averaging the representations); *MaxFusion*, *MinFusion* (taking the maximum/minimum activation across modalities).

Further, we incorporate an additional learnable fusion token (FT) that is added to the input. It is used to aggregate information for the utterance in a similar way as the CLS token is used in BERT, for example. They are labeled as: *SingleFT* (a single FT that is used as a result after encoding); *MainSpeakerFT* (a different FT for each main speaker and one FT for other speakers; *AllSpeakerFT* (incorporating FT for each speaker).

Other possible fusion strategies exist (e.g. Nagrani et al. (2021)), and incorporating some of them is our potential future work.

3.5 Emotion Prediction

Emotions are predicted directly from the fused utterance representations using a two-layer perceptron with LeakyReLU activation function. The hidden size matches the fusion size.

3.6 Link Prediction

The link prediction module is also inspired by the transformer-based encoder. The fused utterance

representations are used as the input tokens. However, our focus shifts from encoded tokens to attention matrices in this context. These matrices are processed and utilized as the adjacency matrix of the graph. Positional encoding is optional since it may be included in the utterance embedding provided by the text, audio, or video encoder.

We used six transformer-based encoder layers, followed by a specialized layer that computes the attention matrices for each head. Subsequently, these matrices are aggregated across all heads using *average*, *maximum*, or *minimum* activation. In contrast to the transformer-based encoder layer, the specialized layer also does not normalize the attention scores, as they are directly provided as logits for the links.

3.7 Text Span Analysis

This section covers the approach used for *Subtask 1*: the prediction of text spans responsible for the cause of emotion. A baseline approach with which we compare is using the entire utterance as a text span. As expected, this trivial approach results in quite a low *strict match F1* metric. The main evaluation metric for this subtask, though, is the *Proportional F1* (which considers the overlap proportion of the predicted span and the annotated one). If we uploaded the text-spans result based on this trivial approach, the resulting *Proportional F1* value is around 20% based on test data provided for the competition evaluation.

Based on this result, we can state that a significant part of training data has no specific text span that causes emotions. This might indicate that even for human annotators, it is not easy to determine a particular text span in a significant number of utterances, and he/she labeled the whole utterance.

According to the training data, we specified five text span categories and created a classifier for their prediction based solely on individual utterances with no context. Furthermore, we have defined a set of regular expressions whose goal is to automatically detect individual categories.

The most common label is `Whole Utterance`, as we declared above. Figure 3 shows all categories of text spans resulting from regular expressions. Regular expressions are used to split an utterance by the punctuation marks `(',', ';', '.', '!', '?')` and compare their results with annotated text spans.

The `First part` label corresponds to the beginning of an utterance until the first punctuation mark. In a similar way, the `Last part` category is taken, except that it is taken from the end. The `Middle part` label is the part in the middle (an utterance part without the beginning and end). It appears usually in cases where an utterance is long. For cases when all regular expressions fail (a text span is neither the whole utterance nor the first, last, or middle part), we created a category `Other`. As illustrated in Figure 3, this category is quite common in training data.

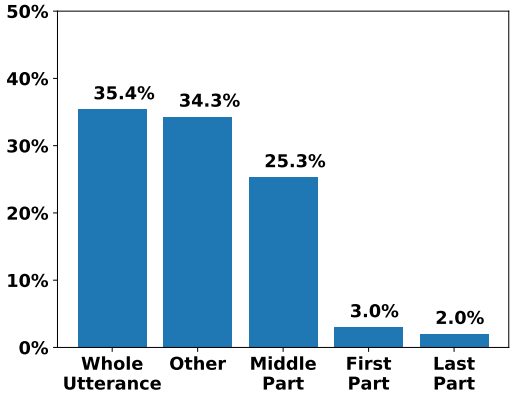


Figure 3: Distribution of the specified categories of text spans

3.7.1 Text Span Classifier

Our first intention was to use a transformer-based model and carry out the question-answering training scenario that aims at providing an "answer" (text span which causes an emotion) identified using the start span and end span generated by the model (similarly to the BERT SQuAD model Devlin et al. (2018)). All our efforts for training such a model, though, failed, due to the lack of training data. The competition rules forbid the usage of another annotated private/public data to fine-tune a model, so we decided to use another model with a much simpler learning objective.

The input text comprises tokens of the current utterance (no context is considered in this case, i.e. no information about previous/subsequent utterances in a dialogue) and is fed into a transformer model as usual with a prediction head with five output neurons.

Once the class (text span category) is predicted from the CLS token, we apply a regular expression (assigned to the predicted class) to extract a substring from the utterance and, in the sequel, the start and end index of this substring.

During the prediction phase, the class `Other` is

taken as `Whole Utterance` label since there is no regular expression associated with this class. We remind that the text span extraction task is solely text-based. The results are presented in Table 4.

4 Experimental Set-up

In the data provided, there is a mapping of video names on train/dev/test splits. According to this information, we dedicated 9,966 utterances (1,373 dialogues) for training, and the remaining dialogues/utterances have been used as development (validation) data. Our preliminary experiments, as well as the experiments resulting in the final system, have been evaluated on this development part of the dataset.

Due to the memory limitations, we fine-tuned audio and video encoders separately. After that, the whole pipeline is trained End-to-End with frozen audio/video encoders and, therefore, constant audio/video representations. We made this choice based on the preliminary experiments, where we obtained the best results with textual modality so we take audio/video features as an auxiliary input.

If not stated differently, we used the AdamW optimizer with a learning rate of 5e-5. Categorical and binary cross-entropy loss was used for emotion and link prediction, respectively. We started with 2 "warmup" epochs with frozen encoders to limit "forgetting" of the pre-trained knowledge. Further, we continued with 50+ epochs until convergence.

To increase the importance of positive links, the positive link weight is set to 5. The fusion size is set to 1024 or 1536. The batch size ranges from 2 to 24. Due to memory limitations, we adapted gradient accumulation technique. The number of samples used for weight update is 8, 12, or 24. These hyper-parameters are further studied in Appendix A. In Tables 1, 2, and 3, we report the combination of these hyper-parameters that obtained the best result among runs.

4.1 Text Encoder

We employed several Pre-trained Language Models (PLMs) and corresponding configurations (see e.g. Table 3). The learning rate for the text encoder was lowered to 1e-7 to further limit the "forgetting" of the PLM.

As described in Section 3, the input is the whole dialogue text to provide context. We set the maximum input length to 450 tokens and the maximum number of utterances to 26, according to the training part of the dataset. That condition is not met in one dialogue in the test part. In that case, the predictions are done in a sliding window manner, so it is impossible to predict the link between the first and last utterance, for example.

For the comparison, we encoded the utterances separately with no dialogue context. This scenario is depicted as ✂ in *Text* column of Tables 1, 2, and 3.

4.2 Text Spans Classifier

All our models have been trained for 30 epochs, with learning rate=1e-05, AdamW optimizer and cross-entropy loss. We picked the best model (the best epoch), based on the validation accuracy. Results for various models are presented in Table 4

4.3 Evaluation Metrics

As stated in the Semeval task information web page[2], the evaluation is based on F1 scores with the help of which we can evaluate the emotion-cause pairs of each emotion category separately and further calculate a weighted average of F1 scores (*wF1*) across the six emotion categories (*Anger*, *Disgust*, *Fear*, *Joy*, *Sadness* and *Surprise*). It is the main evaluation metric for Subtask 2.

Besides the official Semeval metrics provided by the organizers, we have also employed other metrics such as *accuracy* and *macro F1* score for emotion classification task. The *jaccard index* was used for link prediction, since the adjacency matrix is sparse and, therefore, we are not very interested in true negatives.

For Subtask 1 which involves the textual cause span, two strategies are adopted to determine whether the span is extracted correctly: *strict match* (the predicted span should be exactly the same as the annotated span) and *proportional match* (considering the overlap proportion between the predicted span and the annotated one). Although at the beginning of the competition, the main evaluation metric had been *strict match*, later *proportional match* was chosen instead due to the poor results of *strict match* based on trial data published by the organizers. The main reason behind this is that it is challenging to determine the precise boundaries of cause spans.

[2] https://nustm.github.io/SemEval-2024_ECAC/

Text PLM	Text	Audio	Video	Fusion	Multi-task	Acc.	Macro F1
j-hartmann/emotion-english-roberta-large	✓	✗	✗	–	✗	0.859	0.511
bert-large-cased	✓	✗	✗	–	✗	0.859	0.509
bert-base-cased	✂	MFCC	✓	SingleFT	✓	0.852	0.504
bert-base-cased	✓	✗	✗	–	✗	0.857	0.501
bert-large-cased	✓	MFCC	✓	MainSpeakerFT	✓	0.845	0.499
j-hartmann/emotion-english-roberta-large	✓	Wav2Vec	✗	MainSpeakerFT	✗	0.856	0.498
dbmdz/bert-large-cased-finetuned-conll03-english	✓	MFCC	✓	MainSpeakerFT	✓	0.847	0.470
j-hartmann/emotion-english-distilroberta-base	✓	MFCC	✓	MainSpeakerFT	✓	0.837	0.464
–	✗	Wav2Vec	✗	–	✗	0.533	0.397
–	✗	MFCC	✓	SingleFT	✗	0.789	0.296

Table 1: Comparison of emotion prediction methods employing various modalities, fusion scenarios, and combined multi-task training. ✂ denotes separately encoded utterance text.

Text PLM	Text	Audio	Video	Fusion	Multi-task	Jaccard
bert-base-cased	✓	MFCC	✓	MinFusion	✓	0.359
bert-base-cased	✓	Wav2Vec	✗	MaxFusion	✓	0.359
bert-base-cased	✓	✗	✗	–	✓	0.346
bert-large-cased	✓	MFCC	✓	MinFusion	✓	0.342
dbmdz/bert-large-cased-finetuned-conll03-english	✓	MFCC	✓	MainSpeakerFT	✓	0.337
bert-large-cased	✓	MFCC	✓	MainSpeakerFT	✗	0.336
j-hartmann/emotion-english-roberta-large	✓	MFCC	✓	MinFusion	✓	0.331
j-hartmann/emotion-english-distilroberta-base	✓	MFCC	✓	MainSpeakerFT	✓	0.320
bert-base-cased	✂	MFCC	✓	SingleFT	✓	0.279
–	✗	MFCC	✓	SingleFT	✓	0.076

Table 2: Comparison of link prediction methods employing various modalities, fusion scenarios, and combined multi-task training. ✂ denotes separately encoded utterance text.

5 Results

In this section, we present and analyse the results from multiple perspectives.

5.1 Emotion Detection

According to the results presented in Table 1, text plays a crucial role in the emotion prediction task. The context (whole dialogue) is not essential for emotion predictions, as the results are not significantly influenced positively or negatively. We obtained very similar results regardless of whether we used utterances separately (denoted by the symbol ✂) or not.

The best model for the emotion prediction task does not include audio/video features, leading us to conclude that they are not essential for the emotion detection task. The multimodal results suggest that the most effective fusion strategy involves the utilization of the fusion token (*SingleFT* or *MainSpeakerFT*). Other fusion strategies generally yield inferior results.

To support our emotion detection results, we have created a confusion matrix for further error analysis (see Figure 4). The first column (predicted label: `neutral`) indicates that the model has tendencies to predict *neutral* label more often, probably due to the fact that this label is most common in training data. The most challenging emotions are *fear* and *disgust* (see the third and last rows).

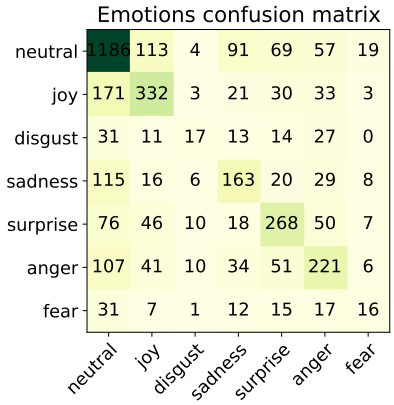


Figure 4: Confusion matrix for the emotion prediction using the submitted model – true labels are at y-axis, predicted labels are at x-axis

Since the overall score is calculated for the predicted triplet: a cause, a target utterance, and an emotion of the target utterance, we estimate the effect of emotion detection accuracy for the ECPE task by comparing the final results with the results of emotions loaded from ground truth (GT). The weighted F1 score on the dev dataset increases from

Text PLM	Text	Audio	Video	Fusion	wF1
bert-base-cased	✓	✗	✗	–	0.320
bert-base-cased	✓	MFCC	✓	MaxFusion	0.318
bert-base-cased	✓	Wav2Vec	✗	MinFusion	0.313
bert-base-cased	✓	MFCC	✓	MinFusion	0.311
bert-large-cased	✓	MFCC	✓	SingleFT	0.310
j-hartmann/emotion-english-roberta-large	✓	MFCC	✓	MinFusion	0.294
dbmdz/bert-large-cased-finetuned-conll03-english	✓	MFCC	✓	MainSpeakerFT	0.289
j-hartmann/emotion-english-distilroberta-base	✓	MFCC	✓	MainSpeakerFT	0.278
bert-base-cased	✂	MFCC	✓	SingleFT	0.262
–	✗	MFCC	✓	SingleFT	0.028

Table 3: Comparison of models trained in multi-task scenario for SemEval task. ✂ stands for separately encoded utterance text.

0.331 to 0.602 if the emotions are correct. Such results suggest that improving the emotion detection model (and increasing the precision and recall) will cause a significantly better overall ECPE score.

5.2 Link Prediction

Our findings demonstrate that injecting emotion information through multi-task training is advantageous for link prediction. The top-performing Jaccard index of 0.359 was attained with multi-task training, while without it, the highest results reached 0.336, marking an improvement of 2.3%.

Results from Table 2 clearly show that the text modality is crucial, as well as the context of the whole dialogue. Using other modalities is helpful in this case.

We have an interesting observation that is contrary to the emotion detection task. In the case of link prediction, it seems beneficial to use fusion strategies based on the aggregation function instead of the fusion token (FT).

5.3 Emotion2Emotion Link Analysis

Our next analysis should shed light on how the model behaves when linking source and target emotions regardless of the utterance texts. We created two matrices (see Figure 5) as follows. We gradually loop through all ground truth and predicted emotion-cause pairs and calculated emotion2emotion pair counts.

Both matrices are very similar (except for the first column, which is automatically zeroed as a part of postprocessing, because ECPE containing *neutral* emotion is irrelevant[3]). It shows that the information about emotions is important for the link prediction task. We supported this observation experimentally, incorporating emotion injection in multi-task training.

The high values in both matrices appear on the diagonal. These represent cases where the source emotion (cause) matches the emotion of the target utterance. The model has evidently learned this behavior, reflecting the human annotations.

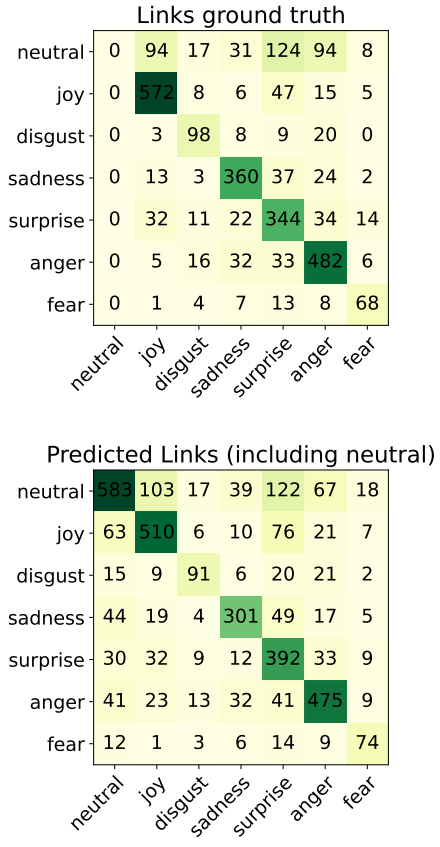


Figure 5: Links in ground truth (top) and predicted links (bottom), emotions loaded from the ground truth – dev dataset

[3]The model creates emotion-cause pairs in *neutral* emotions that should not be possible (no such pairs are present in the training dataset). We remove such pairs before we create the final prediction `json` file.

5.4 Text Span Classification

Table 4 shows the results of various models for the task of text span classification. All models have obtained *accuracy* around 74% – 77% and *Macro-averaged F1 score* above 60%.

Model	Acc.	Macro F1
bert-base-cased	74.1	62.3
bert-large-cased	76.0	60.8
conll03-bert-large-cased[4]	76.6	63.1
roberta-base	**76.8**	61.2
roberta-large	76.7	**67.6**
emotion-roberta-base[5]	76.5	62.4

Table 4: Text span classification results (in %)

For the final submission, we used the bert-large-cased model. However, after the end of the competition, we conducted further experiments with Roberta-like models (see the bottom part of Table 4), and we achieved significantly better scores, particularly in F1 macro.

5.5 Overall Results

For Subtask 2, the main evaluation metric is the weighted F1 score (wF1), as indicated in Table 3 for various models. However, our final submission for the competition is a combination of two models: the best one from emotion detection experiments (Table 1) and the best one for link prediction (Table 2). Such a combination resulted in 0.331 wF1 on dev dataset and 6th place overall in the competition[6]. This is significantly better than the best model from Table 3. All qualitative and error analyses presented above were made based on this setup since test labels are not available at the time being.

For subtask 1, our best model also achieved 6th place in the competition. Our *weighted-avg. proportional F1 score* on test data is 0.208.

Our key findings during the result analysis are as follows:

1. Basic processing of audio/video modalities has brought us only a small positive impact in the case of the link prediction task.

2. The context of the whole dialogue (processing multiple utterances) is crucial for link prediction.

3. We encountered conflicts in fusion strategies (whether to use the aggregation or the fusion token); our best model for emotion prediction is text-only with no fusion mechanism, while the best model for linking benefits from the aggregation fusion strategy.

4. We have obtained better overall results with two separate models.

5. The information about emotion is important for the link prediction task and significantly improves the results.

6 Conclusion

We have participated in two tasks in the *SemEval-2024 Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations* and obtained 6th place overall.

Our model incorporates all modalities (text, audio and video features). The main information lies within the text since the models based solely on textual modality are consistently among the best ones (see Tables 1 – 3). We proposed and implemented several strategies for the fusion of modalities at the utterance level.

To benefit more from video features, we mean that better preprocessing might be helpful (e.g., detecting a main speaker and focusing on her/his face). A possible bottleneck is in the fixed representation of the audio/video features. Optimizing them during the learning process might improve their positive impact and, subsequently, the overall task success rate. Moreover, we can benefit from the usage of a bigger model.

Our experiments have shown that the one multi-task model may not be ideal since optimal hyper-parameters differ for link prediction and emotion detection tasks. Therefore, our final submission is the composition of two models. We have provided a good starting point and a set of analyses for further research.

As a future work, one of our ideas is to use a single learning objective. In such a model, it would not be necessary to have an emotion module since everything would be managed by the link module with a multi-head self-attention matrix where each head would represent a link of one emotion. The training objective should be simpler since it uses single-label classification across components of attention matrices of individual heads. In this way, we can prevent the prediction of neutral links.

[4] dbmdz/bert-large-cased-finetuned-conll03-english

[5] j-hartmann/emotion-english-distilroberta-base

[6] The Semeval official evaluation resulted in 0.251 wF1 on test data

Acknowledgements

This work has been partly supported by the OP JAC project DigiTech no. CZ.02.01.01/00/23_021/0008402 and by the Grant No. SGS-2022-016 Advanced methods of data processing and analysis. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Laura Bostan, Evgeny Kim, and Roman Klinger. 2019. Goodnewseveryone: A corpus of news headlines annotated with emotions, semantic roles, and reader perception. *arXiv preprint arXiv:1912.03184*.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

Xinhong Chen, Qing Li, and Jianping Wang. 2020. A unified sequence labeling model for emotion cause pair extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 208–218.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rob Field. 2022. Speech emotion recognition by fine-tuning wav2vec 2.0. https://huggingface.co/r-f/wav2vec-english-speech-emotion-recognition.

Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Meisd: A multimodal multi-label emotion, intensity and sentiment dialogue dataset for emotion recognition and sentiment analysis in conversations. In *Proceedings of the 28th international conference on computational linguistics*, pages 4441–4453.

Qinghong Gao, Jiannan Hu, Ruifeng Xu, Lin Gui, Yulan He, Kam-Fai Wong, and Qin Lu. 2017. Overview of ntcir-13 eca task. In *NTCIR*.

Lin Gui, Ruifeng Xu, Dongyin Wu, Qin Lu, and Yu Zhou. 2018. Event-driven emotion cause extraction with corpus construction. In *Social Media Content Analysis: Natural Language Processing and Beyond*, pages 145–160. World Scientific.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 45–53.

Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. 2021. Attention bottlenecks for multimodal fusion. *Advances in Neural Information Processing Systems*, 34:14200–14213.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Zafar Rafii and Bryan Pardo. 2012. Repeating pattern extraction technique (repet): A simple method for music/voice separation. *IEEE transactions on audio, speech, and language processing*, 21(1):73–84.

Zafar Rafii and Bryan Pardo. 2013. Online repet-sim for real-time speech enhancement. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 848–852.

Vibha Tiwari. 2010. Mfcc and its applications in speaker recognition. *International journal on emerging technologies*, 1(1):19–22.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2022. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, pages 1–12.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

A Additional Results

Additional results to support the choice of hyper-parameters such as fusion size, positive link weight, and update size, are presented in Tables 5, 6, and 7, respectively.

Fusion size is the hyper-parameter of the Fusion Module (see Section 3). There is a possible pattern in Table 5, that a larger fusion size is beneficial for emotion classification in terms of macro F1 score models. On the other hand, it depends also on the model size. Larger models such as "bert-large-cased" may benefit from larger fusion size, but it seems the opposite for smaller ones such as "bert-base-cased".

Fusion Size	PLM	Jaccard	Acc.	Macro F1	wF1
768	all	0.339	0.845	0.467	–
1024	all	0.338	0.848	0.469	0.295
1536	all	0.336	0.843	0.482	0.293
768	bert-base-cased	0.340	0.847	0.464	–
1024	bert-base-cased	0.339	0.849	0.465	0.294
1536	bert-base-cased	0.325	0.838	0.472	0.277
768	bert-large-cased	0.335	0.844	0.469	–
1024	bert-large-cased	0.338	0.846	0.485	0.305
1536	bert-large-cased	0.342	0.844	0.494	0.302

Table 5: Average results for fusion size hyperparameter across PLMs: Jaccard index is used for link prediction, Accuracy and Macro F1 for emotion prediction, and wF1 for ECPE task

Positive link weight is used in the loss function to increase the importance of positive links in a sparse adjacency matrix. According to Table 6, the hyper-parameter importance is not so significant and the differences are more likely due to other settings such as fusion scenario or PLM. Generally, it worked well with a weight set to 5, which is also the best in terms of macro pair F1 score (*mpF1*).

Weight	Jaccard	Acc.	mpF1
1	0.311	0.836	0.331
5	0.325	0.833	0.345
10	0.319	0.829	0.336
20	0.329	0.834	0.339
50	0.319	0.838	0.335

Table 6: Average results for different weights of positive link: Jaccard index is used for link prediction, Accuracy for emotion prediction, and mpF1 for ECPE task

Update size represents the number of samples used for one weight update using gradient accumulation technique. There is a drop in performance with larger update size in Table 7.

Update Size	Jaccard	Acc.	Macro F1	wF1
8	0.338	0.846	0.484	0.310
12	0.336	0.845	0.480	0.297
24	0.345	0.849	0.469	0.301
60	0.340	0.852	0.462	0.302
120	0.290	0.841	0.414	0.207

Table 7: Average results for different update size (batch · gradient accumulation steps): Jaccard index is used for link prediction, Accuracy and Macro F1 for emotion prediction, and wF1 for ECPE task

GAVx at SemEval-2024 Task 10: Emotion Flip Reasoning via Stacked Instruction Finetuning of LLMs

Vy Nguyen[◦], **Xiuzhen Zhang**[†]
[◦]School of Science, Engineering & Technology, RMIT University
[†]School of Computing Technologies, RMIT University
[◦]s3964786@rmit.edu.vn, [†]xiuzhen.zhang@rmit.edu.au

Abstract

The Emotion Flip Reasoning task at SemEval 2024 aims at identifying the utterance(s) that trigger a speaker to shift from an emotion to another in a multi-party conversation. The spontaneous, informal, and occasionally multilingual dynamics of conversations make the task challenging. In this paper, we propose a supervised stacked instruction-based framework to finetune large language models to tackle this task. Utilising the annotated datasets provided, we curate multiple instruction sets involving chain-of-thoughts, feedback, and self-evaluation instructions, for a multi-step finetuning pipeline. We utilise the self-consistency inference strategy to enhance prediction consistency. Experimental results reveal commendable performance, achieving mean F1 scores of 0.77 and 0.76 for triggers in the Hindi-English and English-only tracks respectively. This led to us earning the second highest ranking[1] in both tracks.

1 Introduction

The EDiReF shared task at SemEval 2024 (Kumar et al., 2024) encompasses two challenges in natural language processing (NLP): *Emotion Recognition in Conversation* (ERC) and *Emotion Flip Reasoning* (EFR). Our work focuses on the latter challenge—EFR, which aims at identifying the utterances responsible for triggering a shift in a speaker's emotional state, hereafter referred to as an *emotion flip*, within a multi-party conversation. The task offers two tracks: one involving Hindi-English code-mixed conversations and the other focusing on English-only conversations. The first track particularly addresses the complexities inherent in multilingual contexts. Each track comes with its respective dataset annotated by the organisers, wherein each utterance is labelled to represent one of the six primary emotional states—*anger*, *disgust*, *fear*, *joy*, *sadness*, and *surprise* (Ekman, 1999). Additionally, the emotion *neutral* is assigned to utterances devoid of any expressed emotion (Kumar et al., 2024). Given these emotion labels, locating the emotion flip is straightforward. Our task is to identify the triggers behind it.

Figure 1 shows a Hindi-English code-mixed conversation conducted between two speakers, complemented by English translations. During the chat, Speaker A undergoes an emotion flip from *sadness* to *joy*, while Speaker B transitions from *surprise* to *joy*. Utterance *u4* is identified as the trigger causing both speakers' emotion flip. Particularly, when Speaker B delivers utterance *u4*, their emotional state also undergoes a change, rendering *u4* as a self-trigger for Speaker B's own emotion flip. It is worth noting that, for an emotion flip, there can be no trigger utterances at all, or there can be one or multiple trigger utterances originating from any involved speakers, including themselves (Kumar et al., 2024).

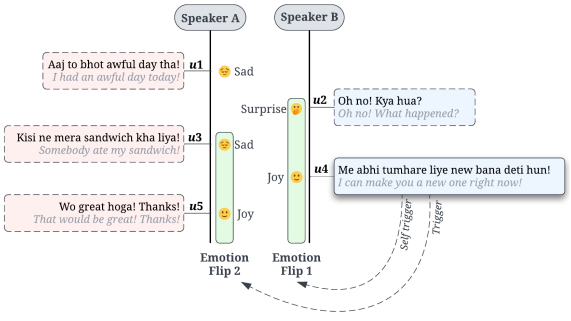


Figure 1: A conversation with five utterances between two speakers involving two emotion flips. Translations are not part of the conversation.

[1]https://codalab.lisn.upsaclay.fr/competitions/16769

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 326–336

The EFR task can be formulated as follows: Given a conversation between p speakers $s_i^{i=1..p}$ involving q utterances $u_j^{j=1..q}$, each assigned an emotion $e_j^{j=1..q}$, if speaker s_i changes their emotion at utterance u_k, there may exist a set of utterances u_l, wherein $1 \leq l \leq k$, that trigger the emotion flip. If we use 1 to denote a *trigger utterance* and 0 to denote a *non-trigger utterance*, then the array $[t_1, t_2, \ldots, t_k]$, in which t_m equals either 0 or 1 and its position in the array corresponds to the position of the utterance in the conversation, can conveniently represent the task's label for an emotion flip. For instance, considering the conversation in Figure 1, the array [0, 0, 0, 1, 0] indicates that the utterance at position 4 caused Speaker A to shift from *sadness* to *joy*.

In this paper, we introduce an instruction-based framework designed to finetune large language models (LLMs) for addressing the EFR task. Initially, leveraging the training data, we construct multiple distinct instruction sets to guide the model in identifying triggers for emotion flips. These instructs emulate human cognitive processes, incorporating both human feedback and self-evaluation procedures as integral components of the reasoning process. Subsequently, we execute a supervised stacked finetuning pipeline to refine the model using these instructions. Once the model is tuned, we employ an inference strategy called *self-consistency* (Wang et al., 2023) to generate predictions for the test data.

Besides the system description, we made the following observations in our experiments:

1. Our framework demonstrates competent performance for both English-only and Hindi-English code-mixed datasets, indicating its capacity to effectively handle both monolingual and multilingual contexts.
2. Providing high-quality instructions to LLMs is crucial for achieving the desired output. Our model's performance improves each time we provide more refined instructions.
3. The self-consistency inference strategy helps mitigate the randomness in the output generated by LLMs, allowing us to attain more uniform results across executions.

In the next section, we discuss various related works. Subsequently, we detail our proposed system in Section 3. Following this, we outline our experimental setup in Section 4, analyse its results in Section 5 before concluding in Section 6.

2 Related Work

The EFR task was first introduced by Kumar et al. (2022), who utilised a masked memory network and a transformer-based architecture to tackle it. In subsequent research in 2023, they delved deeper into the instigators behind emotion flips and introduced a neural architecture named TGIF. This architecture integrates transformer encoders and stacked gated recurrent units (GRUs) to comprehensively capture the conversation context, speaker dynamics, and emotional sequences.

While EFR remains a relatively recent task, it is closely related to the widely studied task of *Emotion-Cause Pair Extraction* (ECPE) (Kumar et al., 2022). The objective of ECPE is to identify a text span that elicit a specific emotion (Xia and Ding, 2019). Earlier endeavours to address ECPE using deep learning faced challenges associated with position bias (Ding and Kejriwal, 2020). Zheng et al. (2022) introduced UECA-Prompt, a BERT-based universal prompt tuning method. Subsequently, Wu et al. (2024) proposed the DECC framework, which incorporates inducing inference and logical pruning to guide LLMs to reason. Both prompt-based approaches outperformed previous works on this task. The promising results observed in ECPE using prompt-based methods motivates us to adapt them to the EFR task.

Prompt-based learning refers to prompting pre-trained language models to tackle downstream tasks (Liu et al., 2021). Recently, LLMs like GPT (OpenAI, 2023) and LLAMA (Touvron et al., 2023) demonstrate exceptional performance across various NLP tasks, even with zero-shot or few-shot prompts (Brown et al., 2020; Sun et al., 2023). Several prompting techniques have emerged recently. Chain-of-thoughts (CoT) prompting, one of the most popular techniques, replicates human cognitive process by integrating intermediate reasoning steps (Wei et al., 2023). Instead of attempting to reach the answer in a single leap, this approach encourages the model to divide complicated problems into smaller, more manageable components, imitating the way humans think. Tree-of-thoughts prompting extends CoT by constructing a tree of logical steps towards the solution (Yao et al., 2023). Multimodal CoT combines text and vision into a two-phase cognitive process (Zhang et al., 2023). On the other hand, instead of fixed prompts, LLMs themselves can be used to dynamically generate prompts for downstream tasks (Zhou et al., 2022) or to produce

and execute programming code as intermediate steps (Gao et al., 2022). Interestingly, LLMs are demonstrated to be capable of generating and analysing recursive reasoning, a unique cognitive ability akin to human thinking processes (Dąbkowski and Beguš, 2023).

Despite these emerging techniques, LLMs often generate outputs that deviate from the ground truth labels (Wadhwa et al., 2023). To address this challenge, instruction tuning emerges as a solution, employing supervised learning on a set of instructions to narrow the gap between the output generated by the base LLMs and the desired output (Zhang et al., 2023). Additionally, human and augmented feedback play a crucial role in mitigating this issue. Akyurek et al. (2023) introduced a reinforcement learning framework equipped with a critique generator to guide GPT-3 in improving its output. Diao et al. (2023) proposed the Active-Prompt method, which entails human manual annotation of uncertain rational chains. Furthermore, Paranjape et al. (2023) devised a novel framework that freezes LLMs and integrates reasoning steps from an external program.

These prior studies underscore the significance of furnishing high-calibre instructions and feedback, as well as employing suitable prompting techniques, to achieve the desired output with LLMs.

3 Our System

In this section, we describe the general approach and the implementation of our system.

3.1 General Approach

Our system must be built upon an *instruction tuneable LLM*. The approach involves two stages: instruction tuning and inference.

3.1.1 Instruction Tuning

Our approach is founded on the premise that problems necessitating reasoning often allows multiple reasoning paths to arrive at the same correct solution. To instil the desired reasoning capabilities in an LLM, we adopt a supervised tuning approach using instructions derived from the training data (Zhang et al., 2023) and implement a stacked framework employing diverse instruction sets to foster the model's ability in navigating varied reasoning paths. A summary of each step is provided below.

Step 1. We train the base model with *Chain-of-thoughts (CoT) instructions*. These instructions can be generated from the training data. This step trains the model on *what is right*.

Step 2. We further provide *feedback-based instructions* to tuned model, expecting it to *rectify the discrepancy* between its current reasoning manner and the desired reasoning manner.

Step 3. We further provide *self-evaluation instructions* to the tuned model, expecting it to enhance its ability to improve itself through *autonomous* evaluation.

Figure 2 summarises the main steps in this supervised finetuning pipeline. Section 3.3.1 describes how we construct these instruction sets.

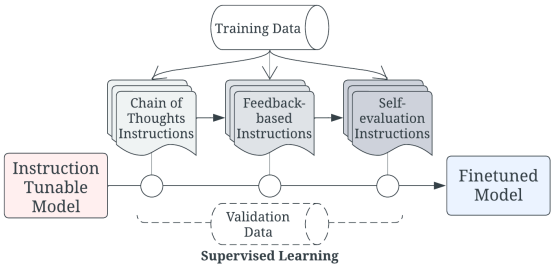


Figure 2: Supervised instruction tuning pipeline.

3.1.2 Inference Procedure

Language models are susceptible to random errors in reasoning, potentially resulting in incorrect conclusions (Wang et al., 2023). To mitigate this issue, these researchers introduced the *self-consistency* (SC) inference strategy. It operates on the principle of generating diverse reasoning paths and selecting the most consistent conclusion by marginalising any inconsistent ones. We adapt this inference strategy to align with the characteristics of our own model.

In our tailored version of SC, we iteratively prompt the model with a progressively increasing *temperature*, a parameter controlling the randomness of the output (Wang et al., 2020), until the answers converge. We introduce a threshold α to determine the convergence point. The convergence condition is if the average of the predicted labels for an utterance is not less than α or not greater than $1-\alpha$. Once the answers converge, the final label for each utterance is the average prediction rounded to the nearest integer, which is eventually either 0.0 or 1.0. Besides the α threshold, we also impose a minimum and maximum number of prompts so that sufficient runs are performed while ensuring the algorithm

still stops if it does not converge. Figure 3 presents a piece of pseudo-code for this inference strategy.

***Global variables*:** *min*, *max*, α

***Converged*(*prediction*):**
If each *item* in *prediction* array ≥ α or ≤ 1 − α:
Return True
Return False

***Infer*(*prediction, dialogue, temp, count*):**
Prompt for *dialogue,* with temperature as *temp*
Assign result array to *new_prediction*
Set *prediction* to mean of all predictions so far
Increase *count* by 1
Increase *temp* by a constant
If (*max* ≤ *count*) or
(*min* ≤ *count* and *Converged*(*prediction*)):
Return *round*(*prediction*)
Return *Infer*(*prediction, dialog, temp, count*)

Figure 3: Our tailored version of SC.

3.2 System Implementation

In this section, we describe how we implement the framework we conceptualise above.

3.2.1 Instruction Construction

We use the training data to construct the instruction sets. Each instruction comprises two components: a *prompt* and a *desired output*. Our finetuning pipeline requires three different instruction sets to be built as follows.

CoT instruction set—The *prompt* includes a labelled conversation sampled from the training data, a CoT that describes the progression of emotional states for the last speaker, and a query tasking the model with identifying the triggers. The *desired output* is a CoT that leads to the accurate identification. We programmatically generate these instructions using a dynamic text template that outlines the sequence of reasoning. The template contains placeholders that can be populated with matching information derived from the conversation. Figure 4 shows how a CoT instruction is crafted for a typical conversation, where each utterance originates from a single speaker, an emotion flip trigger is present, and it is not a self-trigger. Our implementation of the text template is versatile, capable of accommodating various scenarios, including those with no triggers, self-triggers, multiple triggers, and instances where an utterance is attributed to multiple or all speakers.

Feedback-based instruction set—The *prompt* is constructed by sampling a labelled conversation and asking the model to identify the emotion flip triggers directly. Subsequently, its output is then compared with the ground truth labels. If discrepancies arise, the *prompt* is extended with feedback regarding missed or misidentified triggers, and a request for the model to retry the task. We utilise the model tuned using CoT instructions for this step, which enables us to assess its current reasoning manner. Following this, the *desired output* is a CoT that leads to the correct answer. Figure 5 provides an overview of constructing a feedback-based instruction through the integration of a labelled conversation and a baseline model. In our implementation, we again employ dynamic text templates to generate the prompt and desired output for various scenarios, including instances where multiple triggers are missed or misidentified, all triggers are misidentified, and self-triggers are misidentified.

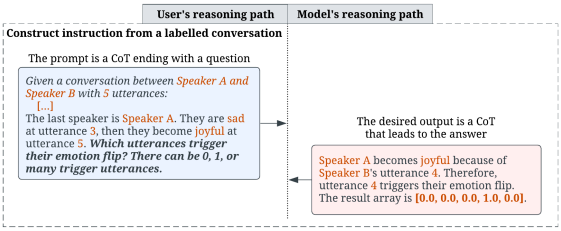


Figure 4: The construction of a CoT instruction for a conversation. Texts in colour indicate placeholders.

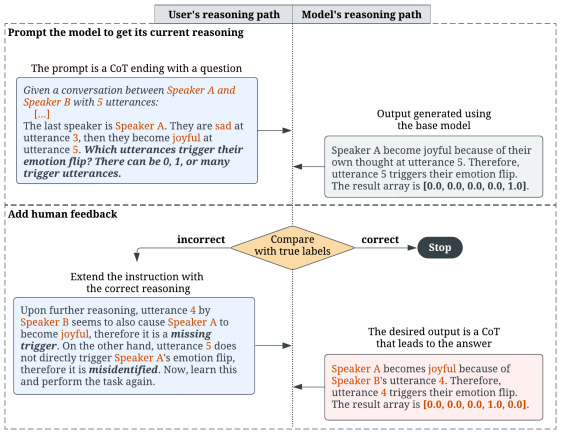


Figure 5: The construction of a feedback-based instruction for a conversation. Texts in colour indicate placeholders.

Self-evaluation instruction set—The *prompt* is structured similarly to a feedback-based instruction, involving the selection of a labelled conversation, and prompting the model finetuned in Step 2 to replicate its current reasoning approach. However, in cases where the output is inaccurate, the prompt extends to instruct the

model to evaluate its own output and iteratively retry the task until satisfaction is achieved. The *desired output* is an augmented CoT that emulates a recursive reasoning and evaluation process, culminating in the correct answer. Our approach to constructing self-evaluation instructions is inspired by research indicating that LLMs possess recursive reasoning abilities (Dąbkowski and Beguš, 2023). Leveraging this capability, we instruct LLMs to engage in autonomous evaluation. To implement this idea, we compile a dynamic text template to simulate a recursive thinking process with information extracted from the given conversation. This template enables the generation of a variable number of iterations, mirroring the iterative cognitive process observed in humans, which may not always yield perfect results in the initial iterations. Figure 6 illustrates the construction of the prompt and the simulation of an expected output using two reasoning iterations before reaching a correct answer.

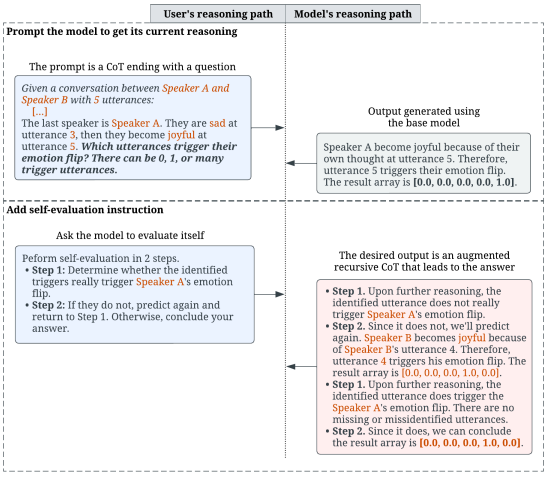


Figure 6: The construction of a self-evaluation instruction for a conversation. Texts in colour indicate placeholders.

As the building of the feedback-based and the self-evaluation instruction sets requires the model to undergo learning from the preceding step, our system must be finetuned in a sequential pipeline.

3.2.2 Prompting Finetuned Model

Following the finetuning of the base LLM with the three prepared instruction sets, we proceed with the SC inference procedure to make predictions for unlabelled data. A critical aspect of this process involves prompting the finetuned model in diverse manners to elicit varied reasoning paths. Given the utilisation of three instruction sets, we employ three distinct prompt variants to prompt the model in identifying emotion flip triggers. The prompt variants utilised are detailed in Figure 7.

Extracting the label from the output sequence generated by the model requires engineering effort due to the dynamics of LLMs. When multiple labels exist in the output, our implementation selects the last label. This aligns with our tuning technique, where intermediate predictions may undergo adjustments during subsequent re-evaluations.

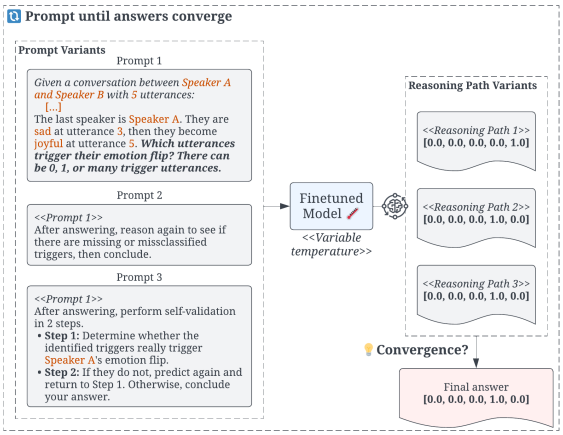


Figure 7: Multiple prompts variants are utilised to produce varied reasoning paths.

4 Experimental Setup

4.1 Datasets

In our experiments, we utilise the datasets provided by the organisers. The data for both tracks originate from previously published datasets. The Hindi-English dataset is sourced from MaSaC, a multimodal dataset compiled from the Hindi TV series *Sarabhai vs Sarabhai* (Bedi et al., 2023). The English monolingual dataset is sourced from MELD, a dataset containing dialogues from the American TV sitcom *Friends* (Poria et al., 2019). In the Hindi-English track, a new emotion, *contempt*, appears, which does not impact our approach since it solely focuses on the positions of the utterance before and after the emotion flip, not the emotions themselves. Table 1 summarises the shape of both datasets.

Upon closer examination of the training splits, it is evident that a significant portion of triggers lies within a proximity of either 1 or 2 utterances preceding the emotion flip. Furthermore, conversations in the Hindi-English dataset exhibit greater length and involve more speakers on average compared to the English-only dataset.

Statistics on the training splits for both tracks are shown in Table 2.

Split	Instances	Utterances	Triggers	% Triggers
Hindi-English dataset				
Train	4,894	98,777	6,542	6.62%
Val	3,89	7,462	434	5.82%
Test	3,85	7,690	461	5.99%
English-only dataset				
Train	4,000	35,000	5,575	15.93%
Val	426	3,522	494	14.03%
Test	1,002	8,642	1,169	13.53%

Table 1: Shape of the datasets provided.

	Utterances	Triggers	Speakers	Distance
Hindi-English dataset				
Min	1.00	0.00	1.00	0.00
Mean	20.19	1.34	3.59	1.43
75%	27.00	2.00	4.00	2.00
Max	106.00	6.00	10.00	21.00
English-only dataset				
Min	2.00	0.00	1.00	0.00
Mean	8.75	1.39	2.62	1.38
75%	12.00	2.00	3.00	1.00
Max	24.00	12.00	8.00	17.00

Table 2: Statistics on the training splits.

4.2 Evaluation Method

We utilise the F1 score of the identified trigger utterances, labelled as 1, as the primary evaluation metric. The F1 score, which balances precision and recall, serves as a robust metric to evaluate the model's ability to accurately identify emotion flip triggers while considering both false positives and false negatives (Goutte and Gaussier, 2005).

To assess the efficacy of our system, we establish a baseline by referring to the results obtained using masked memory networks and transformers by the researchers who proposed the EFR task (Kumar et al. 2022). Subsequently, we conduct an ablation study, aiming to discern the impact of each component in the architecture on the overall performance of the model. Furthermore, we also perform cross-lingual inference to assess the cross-lingual capability of our approach.

4.3 Tuning and Inference Settings

We use the model GPT-3.5-Turbo-1106 by OpenAI[2] as the base model and Azure[3] as the infrastructure. For each track, we separately finetune the model in five epochs using a batch size of 8 and a learning rate multiplier of 1.0, while also incorporating a prompt loss weight. Due to the impromptu and informal nature of conversations, a low content filter setting is consistently used throughout all stages so that the model accepts more contents in their original form.

After finetuning, we generate predictions for the test data using the SC inference strategy. We incorporate a minimum of 3 prompts and a maximum of 10 prompts, alongside an α threshold set at 0.75. This stringent threshold dictates that a consensus of at least 3 out of 4 (75%) agreement amongst predicted labels for an utterance must be achieved before the final label is determined. Furthermore, the temperature parameter is initialised at 0.0 and progressively incremented by 0.1 in each iteration. This iterative adjustment facilitates the introduction of increasing randomness into the model's output, thereby mitigating the risk of overfitting.

5 Results and Analysis

5.1 Main results

In this section, we conduct five executions for each test and report the averages to obtain reliable results. Table 3 provides a summary of the models' performance across all tests conducted.

Initially, to evaluate the base LLMs, we conduct one-shot prompting using the GPT-3.5-Turbo-1106 and GPT-4-0613 models. This prompt construction mirrors that of the CoT instruction set. Our results reveal that GPT-4-0613 achieves an F1 score of 0.60 for the English-only track, surpassing the baseline by 0.061 points, without prior training. Similarly, it shows comparable performance in the Hindi-English track, achieving an F1 score of 0.57.

Subsequent tests demonstrate that finetuning the base GPT-3.5-Turbo-1106 model with additional instructions consistently enhances its performance. We utilise a distinct prompt variant at each tuning stage for zero-shot prediction to prompt the model to reason according to our desired approach, as described in Section 3.3.1. We then apply the SC procedure on the fully tuned model. Integrating all proposed techniques into the final model yields a plateau F1 score of 0.77 and 0.76 for the Hindi-English and English-only tracks respectively. Note

[2] https://platform.openai.com/docs/models

[3] https://azure.microsoft.com

that in the SemEval-2024 Task 10 leader board, we achieved 0.79 for the former track, which was our best run. The results reported in this paper are the mean F1 scores across five runs.

System	Prompt	Accuracy	F1 (0)	F1 (1)
Hindi-English track				
GPT-3.5-Turbo-1106	1-shot	0.95	0.97	0.53
GPT-4-0613	1-shot	0.95	0.97	0.57
GPT-3.5 tuned Step 1	0-shot	0.96	0.98	0.67
GPT-3.5 tuned Step 2	0-shot	0.97	0.98	0.71
GPT-3.5 fully tuned	0-shot	0.97	0.99	0.73
GPT-3.5 fully tuned	SC	0.98	0.99	**0.77**
English-only track				
GPT-3.5-Turbo-1106	1-shot	0.88	0.93	0.57
GPT-4-0613	1-shot	0.89	0.93	0.60
GPT-3.5 tuned Step 1	0-shot	0.91	0.95	0.69
GPT-3.5 tuned Step 2	0-shot	0.92	0.96	0.72
GPT-3.5 fully tuned	0-shot	0.93	0.96	0.74
GPT-3.5 fully tuned	SC	0.95	0.96	**0.76**

Table 3: Model performance in different settings.

5.2 Error Analysis

Our quantitative analysis indicates that the test data provided are representative of the training data. Table 4 shows the confusion matrices of the fully tuned models for both tracks. In the Hindi-English code-mixed track, the model exhibits a tendency to misclassify triggers as non-triggers. Conversely, in the English-only track, a notable balance exists between misidentified triggers and misidentified non-triggers, despite the class imbalance.

Upon closer examination, Table 5 displays the frequency of each type of emotional flip, along with the corresponding number of accurate predictions. In this table, a prediction for a conversation is considered accurate only when all triggers and non-triggers are correctly identified. The data shows that across both tracks, emotion flips from *neutral* to *joy* and from *joy* to *neutral* are the most prevalent. The model achieves accuracy rates of 67.27% and 70.16% in identifying the triggers for these flips in the Hindi-English and English-only tracks respectively.

5.3 Ablation Analysis

In our ablation analysis, we note a consistent improvement in model performance with the addition of each instruction set. Table 6 illustrates these findings, indicating that each successive step reduces the number of false positive and false negative errors from its previous step. Despite that, it also introduces new errors into the predictions; however, the number of new errors is consistently lower than the errors reduced. Notably, tuning the model with CoT instructions at Step 1 emerges as the most impactful, reducing error rates by 38% and 25%, thus increasing F1 scores by 0.15 and 0.12 points for the Hindi-English and English tracks respectively. This highlights the efficacy of instruction tuning. Even with only one instruction set, the disparity between the base model's reasoning manner and the desired reasoning manner is significantly diminished. Subsequent steps further diminish errors, ultimately resulting in the plateau performance observed when employing all techniques in conjunction.

		True Label 0	True Label 1
Predicted	0	**7,201**	113
	1	73	**303**

Hindi-English track

		True Label 0	True Label 1
Predicted	0	**7,197**	285
	1	276	**884**

English-only track

Table 4: Confusion matrices for the fully tuned models.

Hindi-English track

Emotion After \ Emotion Before	*Ag*	*Ct*	*Dg*	*Fr*	*Jy*	*Nt*	*Sn*	*Sp*
Ag		5_3	1_0	1_0	8_7	23_{15}	0_0	3_2
Ct	4_3		0_0	2_0	12_9	15_{10}	0_0	1_1
Dg	3_2	1_1		0_0	0_0	1_1	0_0	1_1
Fr	2_0	0_0	1_0		2_0	13_{11}	2_2	1_1
Jy	6_6	2_1	1_0	3_1		38_{22}	5_4	3_3
Nt	27_{21}	22_{14}	2_0	15_{14}	72_{52}		9_5	13_9
Sn	4_3	2_1	0_0	3_2	12_9	12_6		1_1
Sp	6_6	3_3	0_0	0_0	7_4	14_{14}	0_0	

English-only track

Emotion After \ Emotion Before	*Ag*	*Dg*	*Fr*	*Jy*	*Nt*	*Sn*	*Sp*
Ag		13_7	9_7	14_8	65_{35}	15_{10}	28_{19}
Dg	7_6		1_1	3_2	19_{10}	4_2	5_3
Fr	2_2	1_0		4_2	20_{13}	3_3	4_2
Jy	12_7	1_1	3_3		119_{75}	19_{14}	31_{18}
Nt	73_{55}	16_{13}	17_{14}	119_{92}		47_{40}	67_{54}
Sn	22_{12}	2_0	2_1	13_{10}	49_{32}		17_6
Sp	27_{18}	7_7	2_2	24_{19}	83_{66}	13_{11}	

Table 5: Statistics of the model's accurate predictions for each emotion flip. Cell values present the frequency for an emotion flip, while subscript values present the number of accurate predictions. Top 2 mostly seen flips are highlighted in grey. *Abbreviations:* Anger (Ag), Contempt (Ct), Disgust (Dg), Fear (Fr), Joy (Jy), Neutral (Nt), Sadness (Sn), and Surprise (Sp).

Error	GPT-3.5	+CoT	+Feedback	+Self-eval	+SC
Hindi-English track					
FP	223	113^{-130}_{+20}	105^{-18}_{+8}	89^{-29}_{+13}	73^{-22}_{+6}
FN	185	137^{-61}_{+13}	130^{-14}_{+7}	124^{-10}_{+4}	113^{-17}_{+6}
English-only track					
FP	583	414^{-194}_{+25}	351^{-81}_{+18}	313^{-47}_{+9}	276^{-30}_{+7}
FN	478	344^{-171}_{+37}	319^{-41}_{+16}	298^{-31}_{+10}	285^{-17}_{+4}

Table 6: Ablation analysis of the model performance at each stage. Superscript values indicate the number of errors reduced, while subscript values indicate the number of newly introduced errors. *Abbreviations:* False Positive (FP), False Negative (FN).

5.4 Effectiveness of SC Inference Strategy

Previous sections show that SC improves the F1 score for both tracks. This section proceeds to deep dive into this strategy. Table 7 shows a conversation excerpted from the test data between Mark and Rachel, wherein there exists no triggers for Rachel's emotion flip from *anger* to *neutral*, hence the ground truth label is [0, 0, 0]. This instance is tricky, as Mark's question, Rachel's response, and her subsequent exclamation all appear relevant to the emotion flip. With α set at 0.75, after the first three prompt variants, the model's outputs do not align. However, as we prompt with a progressively higher temperature, convergence is achieved after 8 iterations, with at least 75% of the predictions for each utterance now in agreement. As a result, the predicted label matches the true label. This example aptly illustrates the efficacy of SC in resolving disagreements between different reasoning paths.

Mark: *Why do all your coffee mugs have numbers on the bottom?* [***Surprise***]
Rachel: *Oh. That's so Monica can keep track. That way if one on them is missing, she can be like, "Where's number 27?!"* [***Anger***]
Rachel: *Y'know what?* [***Neutral***]

Iter	Prompt	Temp	Prediction	Running Average
1	1	0.0	[0, 1, 0]	[0.00, 1.00, 0.00]
2	2	0.1	[0, 0, 0]	[0.00, 0.50, 0.00]
3	3	0.2	[0, 0, 1]	[0.00, 0.33, 0.33]
4	1	0.3	[0, 0, 1]	[0.00, 0.25, 0.50]
5	2	0.4	[0, 0, 0]	[0.00, 0.20, 0.40]
6	3	0.5	[1, 0, 0]	[0.17, 0.17, 0.33]
7	1	0.6	[0, 0, 0]	[0.14, 0.14, 0.28]
8	2	0.7	[0, 1, 0]	**[0.14, 0.25, 0.25]**

Table 7: Efficacy of SC in helping resolve disagreements between different reasoning paths for a sample conversation excerpted from test data.

5.5 Cross-lingual Inference

To assess the cross-lingual generalisability of our approach, we use the model trained on the Hindi-English dataset to predict outcomes for the English-only track, and reciprocally, the model trained on the English-only track for the Hindi-English dataset. The results presented in Table 8 demonstrate that our models achieve commendable performance, both surpassing GPT-4-0613, despite not being finetuned on data representative of the test data provided.

Model	Test Data	Accuracy	F1 (0)	F1 (1)
Hindi-English	English-only	0.92	0.95	**0.69**
English-only	Hindi-English	0.96	0.98	**0.64**

Table 8: Model performance using cross-lingual inference.

5.6 Model Hallucination

When fine-tuning GPT-3.5, we encountered a peculiar form of hallucination—an instance where the model generates outputs that largely deviate from the provided training data (Ji et al., 2023). Despite being explicitly instructed to classify *each utterance* as '0' or '1', the model predictions include '2' for some utterances in one execution and include more labels than the number of utterances in another execution. We eventually decided to omit these anomalous executions to maintain the integrity of our results. Currently, controlling this type of hallucination remains a challenge. Further research is necessary to mitigate this phenomenon and improve the model's adherence to its operational constraints.

5.7 Other Constraints

In our experiments, we leverage GPT models hosted on Azure cloud infrastructure. While this offers convenience and efficiency, they are not without their associated costs. Our finetuning process demands 3.5 hours of training time, encompassing approximately 2 million training tokens alongside nearly 200,000 prompting tokens. Additionally, the SC strategy necessitates multiple prompts to attain convergence, thereby extending the time required to derive final predictions. With our settings, the average model speed is 1.22 seconds per prompt for the Hindi-English track and 0.83 seconds per prompt for the English-only track. In light of these considerations, it is crucial to

diligently address cost and resource constraints when building the models.

6 Conclusion & Future Work

The paper presents an instruction based LLM finetuning framework to address the EFR task. Our strategy employs a multilayered finetuning pipeline, utilising three diverse instruction sets to steer the model towards recognising emotion flip triggers and, and finalised with the application of the SC inference strategy. The framework benefits significantly from the provision of high-quality instructions, as evidenced by the progressively improved performance of our model as better-quality feedback and instructions are incorporated into the finetuning pipeline. The robustness of our framework is demonstrated by its proficient handling of both English-only and Hindi-English code-mixed datasets, affirming its effectiveness in varied linguistic scenarios. Through these findings, we trust that our study makes a meaningful impact on the field of prompt-based learning techniques by harnessing the evolved capabilities of LLMs.

Moving forward, our focus will be on an in-depth exploration of various instruction types to devise the optimal way to amalgamate them for the most generalisability. Furthermore, we plan to develop a systematic method for constructing a processing pipeline tailored to this task and potentially applicable to related NLP tasks. This pipeline will be designed to encompass a CoT prompts, incorporate feedback mechanisms, and integrate self-evaluation instructions to ensure a robust, repeatable process for enhancing model performance.

References

Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: Generating Natural Language Feedback with Reinforcement Learning for Repairing Model Outputs. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada. Association for Computational Linguistics.

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Multi-Modal Sarcasm Detection and Humor Classification in Code-Mixed Conversations. *IEEE Transactions on Affective Computing*, 14(02):1363–1375.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, et al. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs].

Maksymilian Dąbkowski and Gašper Beguš. 2023. Large language models and (non-)linguistic recursion. arXiv:2306.07195 [cs].

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active Prompting with Chain-of-Thought for Large Language Models. arXiv:2302.12246 [cs].

Jiayuan Ding and Mayank Kejriwal. 2020. An Experimental Study of The Effects of Position Bias on Emotion Cause Extraction. arXiv:2007.15066 [cs].

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. End-to-End Emotion-Cause Pair Extraction based on Sliding Window Multi-Label Learning. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3574–3583, Online. Association for Computational Linguistics.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6(3–4):169–200.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided Language Models. arXiv:2211.10435 [cs].

Qingqing Gao, Biwei Cao, Xin Guan, Tianyun Gu, Xing Bao, Junyan Wu, Bo Liu, and Jiuxin Cao. 2022. Emotion recognition in conversations with emotion shift detection based on multi-task learning. *Knowledge-Based Systems*, 248:108861.

Cyril Goutte and Eric Gaussier. 2005. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In David E. Losada and Juan M. Fernández-Luna, editors, *Advances in Information Retrieval*, pages 345–359, Berlin, Heidelberg. Springer.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. ACM Computing Surveys, 55(12):248:1-248:38.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Emotion Flip Reasoning in Multiparty Conversations. *IEEE Transactions on Artificial Intelligence*:1–10.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. SemEval 2024 -- Task 10: Emotion Discovery and Reasoning its Flip in Conversation (EDiReF). arXiv:2402.18944 [cs].

Wei Li, Yang Li, Vlad Pandelea, Mengshi Ge, Luyao Zhu, and Erik Cambria. 2023. ECPEC: Emotion-Cause Pair Extraction in Conversations. *IEEE Transactions on Affective Computing*, 14(3):1754–

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. arXiv:2107.13586 [cs].

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, et al. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs].

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. ART: Automatic multi-step reasoning and tool-use for large language models. arXiv:2303.09014 [cs].

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Xiaofei Sun, Linfeng Dong, Xiaoya Li, Zhen Wan, Shuhe Wang, Tianwei Zhang, Jiwei Li, Fei Cheng, Lingjuan Lyu, Fei Wu, and Guoyin Wang. 2023. Pushing the Limits of ChatGPT on NLP Tasks. arXiv:2306.09719 [cs].

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs].

Somin Wadhwa, Silvio Amir, and Byron C. Wallace. 2023. Revisiting Relation Extraction in the era of Large Language Models. arXiv:2305.05003 [cs].

Juntao Wang and Tsunenori Mine. 2023. Multi-Task Learning for Emotion Recognition in Conversation with Emotion Shift. In Chu-Ren Huang, Yasunari Harada, Jong-Bok Kim, Si Chen, Yu-Yin Hsu, Emmanuele Chersoni, Pranav A, Winnie Huiheng Zeng, Bo Peng, Yuxi Li, and Junlin Li, editors, *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 257–266, Hong Kong, China. Association for Computational Linguistics.

Pei-Hsin Wang, Sheng-Iou Hsieh, Shih-Chieh Chang, Yu-Ting Chen, Jia-Yu Pan, Wei Wei, and Da-Chang Juan. 2020. Contextual Temperature for Language Modeling. arXiv:2012.13575 [cs].

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs].

Jialiang Wu, Yi Shen, Ziheng Zhang, and Longjun Cai. 2024. Enhancing Large Language Model with Decomposed Reasoning for Emotion Cause Pair Extraction. arXiv:2401.17716 [cs].

Rui Xia and Zixiang Ding. 2019. Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arXiv:2305.10601 [cs].

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023a. Instruction Tuning for Large Language Models: A Survey. arXiv:2308.10792 [cs].

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023b. Multimodal Chain-of-Thought Reasoning in Language Models. arXiv:2302.00923 [cs].

Xiaopeng Zheng, Zhiyue Liu, Zizhen Zhang, Zhaoyang Wang, and Jiahai Wang. 2022. UECA-

Prompt: Universal Prompt for Emotion Cause Analysis. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, et al., editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7031–7041, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large Language Models Are Human-Level Prompt Engineers. arXiv:2211.01910 [cs].

NLP_STR_teamS at SemEval-2024 Task1: Semantic Textual Relatedness based on MASK Prediction and BERT Model

Lianshuang Su
School of Information
Science and Engineering,
Yunnan University
su_sls@163.com

Xiaobing Zhou
School of Information
Science and Engineering,
Yunnan University
zhouxb@ynu.edu.cn

Abstract

This paper describes our participation in the SemEval-2024 Task 1, "Semantic Textual Relatedness for African and Asian Languages ." This task detects the degree of semantic relatedness between pairs of sentences. Our approach is to take out the sentence pairs of each instance to construct a new sentence as the prompt template, use MASK to predict the correlation between the two sentences, use the BERT pretraining model to process and calculate the text sequence, and use the synonym replacement method in text data augmentation to expand the size of the data set. We participate in English in track A, which uses a supervised approach, and the Spearman Correlation on the test set is 0.809.

1 Introduction

We participated in the English language of track A in Task 1, "Semantic Textual Relatedness for African and Asian Languages." Track A uses a supervised approach where systems are trained on labeled training datasets. This task detects the degree of semantic relatedness between pairs of sentences for African and Asian Languages (Ousidhoum et al., 2024b).

Semantic Textual Relatedness (STR) is an important measure of the relationship between texts. It is considered to be the basis for understanding meaning(Miller and Charles, 1991) and is crucial for many natural language processing tasks. By computing semantic textual relatedness, we can perform applications such as text matching(Xu et al., 2013), information retrieval(Wagh and Kolhe, 2011), text categorization(Alsamurai, 2017), and question answering systems(Das and Saha, 2022).

However, previous NLP work has focused on semantic similarity (a small subset of semantic relatedness), in large part due to the lack of datasets on relatedness. For example, SemEval-2015 task1 is paraphrase and semantic similarity in twitter(Xu et al., 2015). And SemEval-2016 task1 is semantic textual similarity, monolingual and cross-lingual evaluation(Agirre et al., 2016).

Semantic relatedness and semantic similarity are two ways to explore the closeness of meaning. Two terms are considered semantically similar if there is a synonym, contextual, or modal relation relationship between them. Two terms are considered semantically related if there is any lexical semantic relation between them. Thus, all similar pairs are also related, but not all related pairs are similar(Abdalla et al., 2021). In semantic textual relatedness, we focus on the meaning and semantic information of the text, not just the surface word or sentence structure. Thus, the semantic relatedness between two texts can relate to their themes, intentions, emotions, etc.

The semantic relatedness of texts can be computed using the content and links of hypertext encyclopedias(Yazdani and Popescu-Belis, 2013). Semantic relatedness between texts can also be measured by calculating the similarity between text representations using a pre-trained language model.

In the following, we describe in detail the methods we used and give the evaluation results and conclusions.

2 Background

In this section, we present important details about the task setup. Each instance in the train set, dev set, and test set is a sentence pair, and these two sentences are separated by a newline character. The instance is labeled with a score representing the degree of semantic textual relatedness between the two sentences(Ousidhoum et al., 2024a). As shown in Table 1, there are two sentence pairs examples to present the semantic textual relatedness.

The scores can range from 0 (maximally unrelated) to 1 (maximally related), which are obtained using a comparative annotation framework. The

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 337–341

sentence1	sentence2	STR score
A girl is communicating with sign language.	A young girl is using sign language.	0.83
You should have respect for your mother.	Even if this is your own mother!	0.41

Table 1: Sentence pairs examples

	Train	Dev	Test
before text data augmentation	5500	250	2600
after text data augmentation	11000	250	2600

Table 2: Size of the data set

train and dev sets give sentence pairs and semantic textual relatedness scores, and the test set only gives sentence pairs. The train set was enlarged by using text data augmentation. The size of the dataset is shown in Table 2. The task we participated in was the English in track A. The task is a regression task whose input is a sentence pair and the output is the semantic textual relatedness score for that sentence pair.

3 System Overview

In this section, we present our approach applied to the task of predicting STR. We use the BERT pre-training model(Devlin et al., 2018) for text sequence processing and computation, and also employ text data augmentation to improve the training results. We adopted prompt tuning(Liu et al., 2023) to construct a new sentence, "The correlation of the next two sentences sent0 and sent1 is [MASK].", and used this constructed new sentence as a prompt template, where [MASK] is used to predict the correlation between the two sentences.

3.1 Model

We use the BERT pre-training model, designed to pre-train deep bidirectional representations from the unlabeled text by joint conditioning on both the left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks. STR tasks are related to the semantics of the text, so using the case-insensitive English BERT pre-training model works better than the case-sensitive English BERT pre-training model.

We use the BERT model for encoding and feature extraction of text sequences. The structure of the system is shown in Figure 1(Mutinda et al., 2021). The two special tokens [CLS] and [SEP] are added to the model's input data to convert the text into the format expected by BERT. The forward function accepts a batch as input. It extracts $input\_ids$ and $attn\_mask$ from the batch, where $input\_ids$ is a sequence that converts the input text into a numeric representation acceptable to the model, $attn\_mask$ is a sequence of binary masks used to indicate which tokens are real input and which tokens are padded. Then it encodes the $input\_ids$ and $attn\_mask$ to obtain the $enc\_outputs$, which are hidden states of the model's output. Next, the corresponding embedding representation is extracted from the hidden state based on the mask position. These embedding representations are processed through a linear transformation to end up with a scalar value $logits$. Sigmoid activation is performed on $logits$ to get the output score.

3.2 MASK Prediction

Since the labels in the train set are continuous, we modeled this task as a regression problem. We adopted prompt tuning and used the Pattern-Exploiting Training (PET) method(Schick and Schütze, 2021) to construct a new sentence "The correlation of the next two sentences sent0 and sent1 is [MASK]." as a prompt template. In this prompt template, sent0 and sent1 are two sentences, and [MASK] is used to predict the correlation between the two sentences. Thus, it could convert the downstream task into a Complete Fill-in-the-Blank(cloze) task(Ding et al., 2021), and Masked Language Modeling(MLM)(Wettig et al., 2023) BERT can be used for prediction. Since the language of our participation is English, this prompt template is constructed in English. If we want to evaluate the semantic textual relatedness in other languages, we need to modify this template to the corresponding language. The constructed prompt template is fed into the model using the pre-training

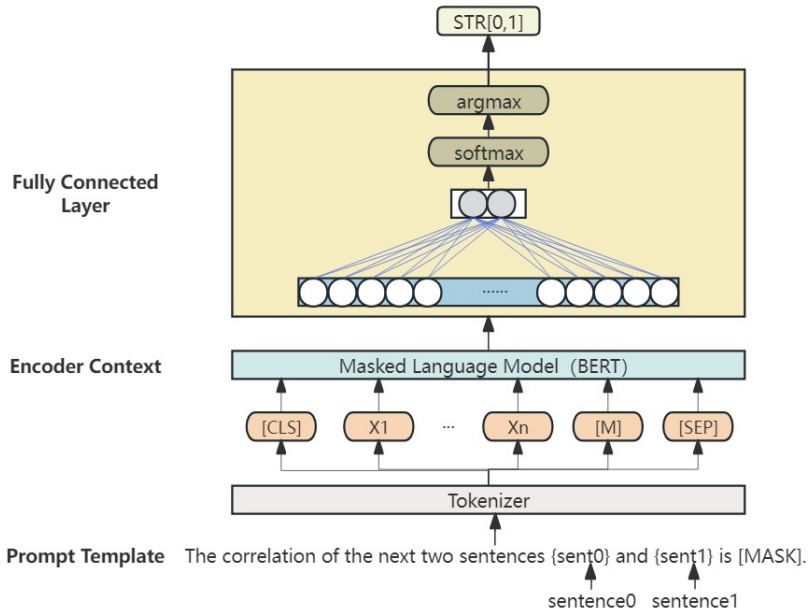


Figure 1: System model structure

model BERT, and the model predicts the representation of the correlation based on the context and the position of [MASK].

3.3 Text Data Augmentation

Through text data augmentation, more training samples can be generated to expand the size of the train set. Besides, text data augmentation can improve the generalization ability and robustness of the model. For example, Connor Shorten and others used a CNN model combined with text data augmentation EDA when the training set size was 5000, and the result improved from 87.7 to 88.3(Shorten et al., 2021). This task is to find out the degree of semantic correlation between two sentences, considering the task requirements and data characteristics, the data samples after performing text data augmentation can change the expression of the sentences, but the overall semantics of the sentences should remain unchanged.

Therefore, we used the synonym replacement method in the text data augmentation method in our experiments instead of random insertion, deletion, and other methods. After using this method changes the number of samples in the train set from 5500 to 11000.

4 Experimental Setup

The data set is given in CSV file format by the SemEval 2024 shared task organizer. It has three columns: PairID, Text, and Score, where Text is a sentence pair. We take out the two sentences in the sentence pair and use these two sentences to construct a new sentence: "The correlation of the next two sentences sent0 and sent1 is [MASK].". This new sentence is then fed into the model for processing and training. When performing text data augmentation, we replace the two sentences with synonyms and then insert the newline character in the middle of the replaced sentence pairs to ensure that the data format is consistent with the original data set.

We use the BERT pre-training model to process and calculate text sequences. The text data augmentation method is synonym replacement. Since this task is a regression task, we use the mean squared error(MSE) loss function:

$$L_{mse} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{'})^2 \quad (1)$$

where y_i is the ground truth for sample i, $y_i^{'}$ is the prediction score for sample i, n is the number of samples.

The batch size is set to 64, the number of training iterations is 6, and the learning rate is 2e-2. At the same time, in order to help the model converge better and achieve better performance, we set up a learning rate scheduler.

	Dev Score	Test Score
before text data augmentation	0.819	0.820
after text data augmentation	0.832	0.809

Table 3: Score on the dev set and test set

5 Results and Analysis

5.1 Results

This section shows the results of our system on the English STR task in track A of SemEval-2024 task 1. We use the Spearman correlation between system output and human annotation as an evaluation metric. Under the premise that other conditions are the same, we use the data set after text data augmentation for training. As shown in Table 3, the Spearman Correlation obtained on the dev set increased from 0.819 to 0.832, but the Spearman Correlation obtained on the test set dropped from 0.820 to 0.809.

5.2 Analysis

As shown in the results, after text data augmentation, the Spearman Correlation obtained on the dev set has improved, but the Spearman Correlation obtained on the test set has declined. Because before text data augmentation, the dev set was around 4.5% size of the train set and the test set was around 47% size of the train set. The size gap between the data sets is large. In addition, relying solely on semantic synonym replacement in sentences for data augmentation will have certain inaccuracies which leading to biased estimates. At the same time, text data augmentation doubled the size of the train set, resulting in a larger difference in the size of the train set, dev set, and test set.

6 Conclusion

This paper describes our participation in the SemEval 2024 competition in the Semantic Textual Relatedness for African and Asian Languages task. We participated in the English task in track A. Our approach is to use the BERT pre-training model for text sequence processing and computation, employing text data augmentation to enlarge the size of the train set, and adopting prompt tuning to construct a prompt template "The correlation of the next two sentences sent0 and sent1 is [MASK].", where [MASK] is used to predict the correlation between two sentences. The final Spearman Correlation obtained on the test set was 0.809.

In the future, we will use methods such as context awareness and manual intervention to address errors caused by text data augmentation to ensure their accuracy and rationality. At the same time, we will expand the size of the dev set, reduce the size difference between data sets, and try to use other more powerful pre-trained models.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *arXiv preprint arXiv:2110.04845*.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Ather Abdulrahem Mohammedsaed Alsamurai. 2017. Text categorization based on semantic similarity with word2vector. Master's thesis, Fen Bilimleri Enstitüsü.

Arijit Das and Diganta Saha. 2022. Deep learning based bengali question answering system using semantic textual similarity. *Multimedia Tools and Applications*, 81(1):589–613.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Minjie Ding, Mingang Chen, Wenjie Chen, and Lizhi Cai. 2021. English cloze test based on bert. In *International Conference on Knowledge Science, Engineering and Management*, pages 41–51. Springer.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Faith Wavinya Mutinda, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. 2021. Semantic textual similarity in japanese clinical domain texts using bert. *Methods of Information in Medicine*, 60:e56–e64.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *Preprint*, arXiv:2402.08638.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8:1–34.

Kishor Wagh and Satish Kolhe. 2011. Information retrieval based on semantic similarity using information content. *International Journal of Computer Science Issues (IJCSI)*, 8(4):364.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2977–2992. Association for Computational Linguistics.

Jiaming Xu, Pengcheng Liu, Gaowei Wu, Zhengya Sun, Bo Xu, and Hongwei Hao. 2013. A fast matching method based on semantic similarity for short texts. In *Natural Language Processing and Chinese Computing: Second CCF Conference, NLPCC 2013, Chongqing, China, November 15-19, 2013, Proceedings 2*, pages 299–309. Springer.

Wei Xu, Chris Callison-Burch, and William B Dolan. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 1–11.

Majid Yazdani and Andrei Popescu-Belis. 2013. Computing text semantic relatedness using the contents and links of a hypertext encyclopedia. *Artificial Intelligence*, 194:176–202.

Halu-NLP at SemEval-2024 Task 6: MetaCheckGPT - A Multi-task Hallucination Detection Using LLM Uncertainty and Meta-models

Rahul Mehta*
IIIT Hyderabad, India
rahul.mehta@research.iiit.ac.in

Andrew Hoblitzell*
Purdue University, USA
ahoblitz@purdue.edu

Jack O'Keefe
Northwestern University, USA
jackokeefe2024@u.northwestern.edu

Hyeju Jang
Indiana University Indianapolis, USA
hyejuj@iu.edu

Vasudeva Varma
IIIT Hyderabad, India
vv@iiit.ac.in

Abstract

Hallucinations in large language models (LLMs) have recently become a significant problem. A recent effort in this direction is a shared task at Semeval 2024 Task 6, **SHROOM**, a *Shared-task on Hallucinations and Related Observable Overgeneration Mistakes (Mickus et al., 2024)*. This paper describes our winning solution ranked *1st* and *2nd* in the 2 sub-tasks of model agnostic and model aware tracks respectively. We propose a meta-regressor framework of LLMs for model evaluation and integration that achieves the highest scores on the leader board. We also experiment with various transformer based models and black box methods like ChatGPT, Vectara, and others. In addition, we perform an error analysis comparing GPT4 against our best model which shows the limitations of the former.

1 Introduction

The recent rapid deployment of large language models (LLMs) has led to a hallucination proliferation which poses a barrier to the reliability and trustworthiness of LLMs (Lin et al., 2022). One of the widely agreed upon definition of hallucinations (Maynez et al., 2020; Xiao and Wang, 2021) is output text containing information not relevant to the input or a desired output. Hallucinations should not be thought of as an occasional nuisance, but rather as a systemic issue inherent to these models and their web-sourced training data which can be rife with bias and misinformation. This can directly cause user discontent when these systems are implemented in production or real-world scenarios.

These type of hallucinations have been widely studied in the context of text related tasks like machine translation (Dale et al., 2022; Guerreiro et al., 2023a,b), summarization (Huang et al., 2023; van der Poel et al., 2022) and dialogue generation (Shuster et al., 2021a). Gaps in hallucination detection methods in LLM outputs persist across many such tasks.

Despite some progress in hallucination detection, existing methods may rely upon comparisons to reference texts, overly simplified statistical measures, reliance upon individual models, or annotated datasets which can limit their extensibility. Our approach leverages the uncertainty signals present in a diverse basket of LLMs to detect hallucinations more robustly.

In this paper, we present a meta-regressor framework for LLM model selection, evaluation, and integration.[1] The overall approach is depicted in Figure 1. For the first step, each LLM-generated sentence is compared against stochastically generated responses with no external database as with SelfCheckGPT (Manakul et al., 2023). A meta-model that leverages input from a diverse panel of expert evaluators evaluates and integrates the output of multiple iterations of the process.

Our framework focuses on creating a meta-model for identifying hallucinations, with the idea that the meta-model's prediction power is linked to the performance of the underlying base models. This model achieves the highest scores in the SemEval-2024 Task 6 competition across three subtasks: Machine translation, Paraphrase generation, and Definition modeling.

2 Related Work

In this section, we describe prior work on hallucination detection methods. We will examine two potential streams for hallucination detection: model-aware detection and black-box detection. Model-aware techniques have access to the model's internals, such as weights and logits while black-box methods do not has access to such model internals.

*Equal contribution

[1]The code of MetaCheckGPT is available at https://github.com/rahcode7/semeval-shroom

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 342–348

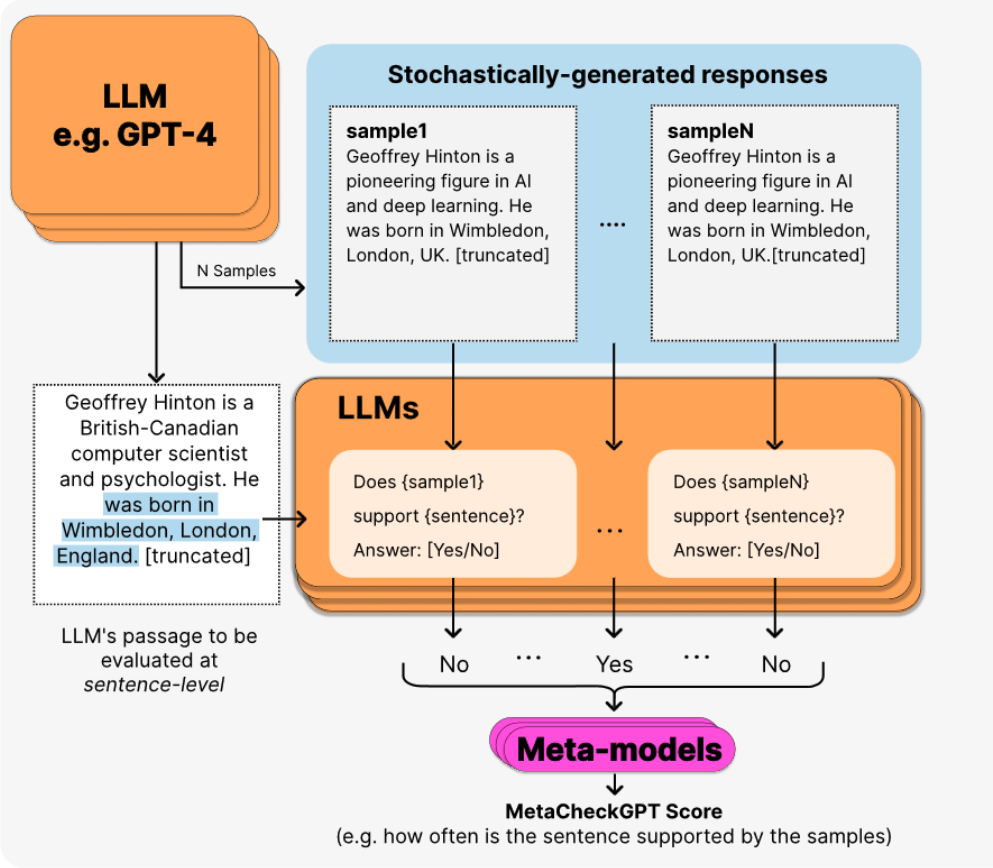


Figure 1: MetaCheckGPT: Generated sentences are compared against stochastically generated responses.

2.1 Model aware Detection

These methods require access to model weights and their logits (van der Poel et al., 2022). For machine translation task, Guerreiro et al. (2023b) showcased that sequence log-probability performs quite well compared to reference based methods. For article generation task, (Varshney et al., 2023) uncertainty estimation techniques(Azaria and Mitchell, 2023) (Tian et al., 2023) were used to detect hallucination in ChatGPT. Other methods to detect hallucinations include Retrieval Augmented Generation (Shuster et al., 2021b) and Chain of Verification based techniques (e.g., (Lei et al., 2023)).

2.2 Black box Detection

With the prevalence of closed source models, there has been recent work on black-box hallucination detection methods which doesn't require the model inputs, only the generated text. For example, a recently proposed system SelfCheckGPT (Manakul et al., 2023) utilizes a sampling-based technique based on the idea that sampled responses for hallucinated sentences will contradict each other.

This model achieves the highest scores across two sub-tasks: Machine translation, Paraphrase generation, and Definition modeling. We perform extensive studies of LLMs like ChatGPT, Mistral, and others to showcase their failure points.

3 Task Description and Datasets

In the SHROOM Task-6, the organizers propose a binary classification task to predict a machine generated sentence is a hallucination or not.

The organizers considered 3 types of text generation tasks - Definition Modelling, Machine Translation and Paraphrase Generation.

3.1 Task Tracks

The shared task was further divided into 2 tracks: **model agnostic** and **model aware**. Figure 2 describes sample examples of hallucinations containing source, reference and output text for each task type.

3.1.1 Model Agnostic Track

In this track, only the inputs, references and outputs were provided The details of the model that produced the text was masked from the participants.

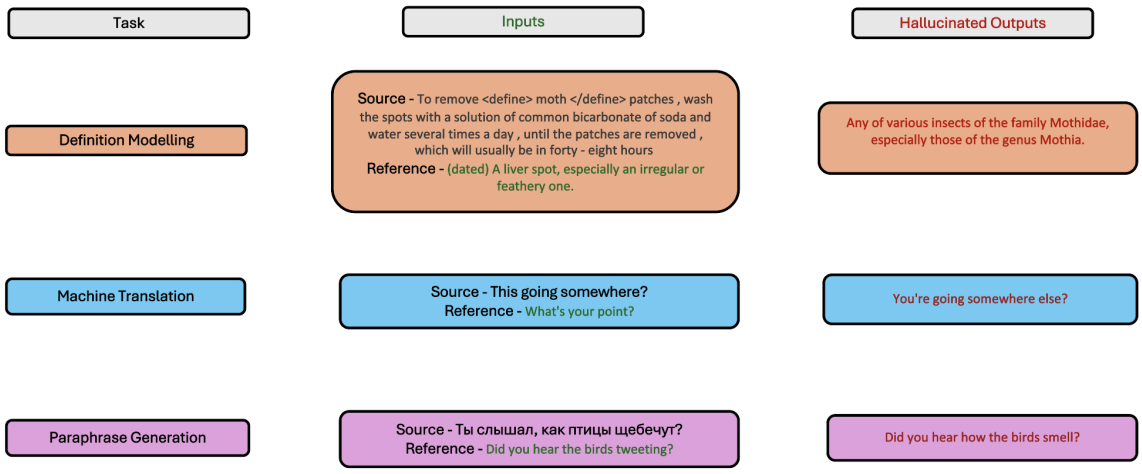


Figure 2: Hallucination examples for each task type

For data preparation, the SHROOM organizers followed the structure described in (Bevilacqua et al., 2020).

	Model Agnostic Track		
Task	Train	Dev	Test
Definition Modeling	10000	187	562
Machine Translation	10000	187	563
Paraphrase Generation	10000	125	375
Total	30000	499	1500

Table 1: Sample counts for the Model Agnostic Track

3.1.2 Model Aware Track

In this track, along with the inputs, references and outputs, the model name and its checkpoints were also provided from which the outputs were generated.

	Model Aware Track		
Task	Train	Dev	Test
Definition Modeling	10000	188	562
Machine Translation	10000	188	563
Paraphrase Generation	10000	125	375
Total	30000	501	1500

Table 2: Sample Statistics for the Model Aware Track

It is worthwhile to note that the organizers chose to share the training which was not labeled and only the development set was labeled.

4 Our Methodology

Algorithm 1 Meta-Model Training/Evaluation

Input: Base models M, Meta-models N, Threshold x
Output: Top performing meta-model
for each base model m in M **do**
    $score_m \leftarrow$ Evaluate m (MAE)
end for
$FilteredMs \leftarrow Models$.filter(MAE $< x$)
for each meta-model n in N **do**
    Train n with $FilteredMs$
    $metaScore_n \leftarrow$ Spearman MAE
end for
$TopMeta \leftarrow$ Meta-model in N with lowest Spearman MAE

Our approach is centered around building a meta-model for hallucination detection, with the hypothesis that the quality of prediction from underlying base models is highly correlated with the meta-model's predictive power. Given a set of base models $M = \{m_1, m_2, ..., m_n\}$ and actual labels $L = \{l_1, l_2, ..., l_n\}$ in the dataset, the Spearman correlation between predicted hallucination scores H and actual labels is given by:

$$\rho_s(H, L) = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where d_i is the difference between the ranks of corresponding elements in H and L.

Our overall process was to identify the meta-model that minimized this mean absolute error

(MAE) function ϵ, where

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)$$

because Spearman correlation was one of the secondary metrics for Task 6 evaluation. Here, Y_i represents the actual Spearman correlation values for hallucination and $\hat{Y}_i$ represents the predicted values. Our overall process is captured in Algorithm 1.

Algorithms 2, 3, and 4 detail how some of our different meta-models were trained. These algorithms follow a unified framework, initiating with the setup of training data and labels, with the ultimate aim of fine-tuning a meta-regressor model. A meta-search cross-validation approach was used to conduct a hyperparameter space for each model's architecture. The process involves iterating over the defined hyperparameter space for each algorithm, fitting the meta-regressor with the training data, and concluding with the identification and preparation of the highest-performing model for deployment. The training process for selecting meta-models is included in the Appendix. RMSE, MAE, and R-squared were used as additional proxies in meta-model evaluation.

Because this problem was assessed with binary classification accuracy, data was classified based on the Spearman correlation coefficient according to:

$$\text{Class} = \begin{cases} \text{'Hallucination'}, & \rho_s > 0.5 \\ \text{'Not Hallucination'}, & \text{otherwise} \end{cases}$$

to convert our regression problem into a binary classification task, simplifying the analysis and interpretation of results. Once converted to a classification problem, the primary metric used for evaluation was accuracy. Precision, Recall, F1 Score, and a confusion matrix were used for secondary evaluation.

5 Experiments & Results

5.1 Experimental set-up

Training was conducted both on cloud using APIs as well as locally on V100/A100 GPUs for faster processing times.

We conducted our initial experiments with simpler base models including DeBERTa (He et al., 2021), DistilBERT (Sanh et al., 2020), RoBERTa (Liu et al., 2019), LLaMA 2 (Touvron et al., 2023), and Mixtral of Experts (Jiang et al., 2024) among others. Preliminary results indicated an accuracy of 0.5 to 0.6, prompting us to continue our search for more performant base models.

Additional analysis indicated our base models ChatGPT(Achiam et al., 2023), SelfCheckGPT(Manakul et al., 2023) and Vectara(Hughes, 2023) showed promising results in initial tests, with accuracy in the range of 0.6 to 0.7. Prompt engineering, self-consistency checks and uncertainty based modeling techniques were used to maximize performance in base models. The training process for more performant meta-models, including random forest and elementary neural ensemble models, can be found in the Appendix.

5.2 Results

Classification performance obtained on the training data, which includes an accuracy of 0.8317, precision of 0.7447, recall of 0.875, and an F1 score of 0.8046.

	Positive	Negative
Positive	TP: 49	FN: 5
Negative	FP: 12	TN: 35

Cross-validation and regularization techniques were applied to increase confidence that the performance observed on the training data would be maintained on test data.

Track	Accuracy	Rho	Rank
Aware	80.6	0.71	1/46
Agnostic	84.7	0.77	2/49

Table 3: Final Modeling results on the test set

5.3 Discussion

Our results, as summarized in Table 3, demonstrate the effectiveness of meta-regressor models in detecting hallucinations across various text generation tasks. One of the key strengths of the approach is that a diverse set of base models is able to better capture a wide range of features indicative of hallucinations than a single model or knowledge base alone may be able to. High performance metrics underline the promise of combining base models/knowledge bases through meta-learning.

Our approach is not without its limitations. The black-box nature of some base models (e.g. GPT4), limits understanding of the internal mechanisms

driving the generation and detection of hallucinations. More detailed limitations of the system and directions for future work are examined in the following section.

5.4 Limitations

There are several limitations to the current work. For example, all language models have inherent limitations such as bias and lack of world grounding. Unfortunately, more recent models such as GPT have also started to function as black box systems. The corpus for training data for base language models is predominantly English. The system also would not readily integrate into a production system without additional effort. The system could also benefit from the ability to learn from feedback. All of the base language models may also suffer from potential safety issues like false confidence and over-reliance, etc.

6 Conclusion

Our meta-model strategy represents a step forward in addressing the challenges of mitigating hallucinations and the importance of a nuanced approach to model selection, evaluation, and integration. The work also acknowledges the need for additional research into more transparent, interpretable, and multilingual models, as well as the integration of external knowledge sources and feedback mechanisms to refine and improve hallucination detection methods. In the future, some areas we would like to work on include utilizing additional multilingual datasets, expand the scope of our work to more set of text generation task, and focus more on white box hallucination detection systems.

While the current system was tested on some machine translation tasks, we believe it could benefit from more work on multilingual datasets. The current system could improve by integrating with external knowledge bases via retrieval augmented generation. The system could also be made more usable by distilling its knowledge into a portable fine-tuned model widely available to others. Another area for potential improvement includes integration of human or agent feedback through reinforcement learning.

Acknowledgements

We would like to thank the Machine Learning Collective (MLC), a group which promotes collaboration and mentorship opportunities being accessible and free for all, for helping connect us as researchers. We would also especially like to thank Timothee Mickus for being very responsive to questions on the listserv. We would also like to thank Elaine Zosa, Raúl Vázquez, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki for organizing the Shared-task on Hallucinations and Related Observable Overgeneration Mistakes, we felt delighted to have a forum to work with others who have a shared interest in language model capabilities. Thanks to Steven Bethard, Ryan Cotterell, Rui Yanfrom, and many others for their work on the template which we adapted from.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, and et al. 2023. Gpt-4 technical report. Technical report, OpenAI.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734*.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

David Dale, Elena Voita, Loïc Barrault, and Marta R. Costa-jussà. 2022. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better.

Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023a. Hallucinations in large multilingual translation models.

Nuno M. Guerreiro, Elena Voita, and André F. T. Martins. 2023b. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2023. The factual inconsistency problem in abstractive text summarization: A survey.

Simon Hughes. 2023. Cut the bull... detecting hallucinations in large language models.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, and et al. Marie-Anne Lachaux. 2024. Mixtral of experts.

Deren Lei, Yaxi Li, Mengya Hu, Mingyu Wang, Vincent Yun, Emily Ching, and Eslam Kamal. 2023. Chain of natural language inference for reducing large language model ungrounded hallucinations. *arXiv*, cs.CL(arXiv:2310.03951).

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021a. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021b. Retrieval augmentation reduces hallucination in conversation. *arXiv*, cs.CL(arXiv:2104.07567).

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. https://doi.org/10.48550/arXiv.2305.14975. ArXiv:2305.14975v2 [cs.CL].

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation.

A Appendix: MR Training Processes

Algorithm 2 MR1 Training: Algorithm 2 outlines the process of training a meta-regressor model with hyperparameters for random forest.

Require: X_{train}, y_{train} ▷ Training data and labels
Ensure: $model_{best}$ ▷ Optimally tuned model

$MR \leftarrow MetaRegressor()$
$H \leftarrow \{n\_estimators \in \{\alpha_1, \ldots, \alpha_N\},$
  $max\_depth \in \{\beta_1, \ldots, \beta_M\},$
  $min\_samples\_split \in \{\gamma_1, \ldots, \gamma_L\},$
  $min\_samples\_leaf \in \{\delta_1, \ldots, \delta_K\},$
  $max\_features \in \{$'auto', 'sqrt'$\},$
  $bootstrap \in \{$True, False$\}\}$
$MetaCV = MetaSearchCV(MR, H, cv)$
$MetaCV.fit(X_{train}, y_{train})$
$params_{best} = MetaCV.best_params$
$model_{best} = MetaRegressor(params_{best})$
$model_{best}.fit(X_{train}, y_{train})$

Algorithm 3 MR2 Training: Algorithm 3 outlines the process of training a meta-regressor model with hyperparameters for gradient boosted trees.

Require: X_{train}, y_{train} ▷ Training data and labels
Ensure: $model_{best}$ ▷ Optimally tuned model

$MR \leftarrow MetaRegressor()$
$H \leftarrow \{n\_estimators \in \eta_1, \ldots, \eta_n,$
  $learning\_rate \in \theta_1, \ldots, \theta_n,$
  $max\_depth \in \iota_1, \ldots, \iota_n,$
  $min\_child\_weight \in \kappa_1, \ldots, \kappa_n,$
  $gamma \in \lambda_1, \ldots, \lambda_n,$
  $subsample \in \mu_1, \ldots, \mu_n,$
  $colsample\_bytree \in \nu_1, \ldots, \nu_n,$
  $reg\_alpha \in \xi_1, \ldots, \xi_n,$
  $reg\_lambda \in \zeta_1, \ldots, \zeta_n\}$
$MetaCV = MetaSearchCV(MR, H, cv)$
$MetaCV.fit(X_{train}, y_{train})$
$params_{best} = MetaCV.best_params$
$model_{best} = MetaRegressor(params_{best})$
$model_{best}.fit(X_{train}, y_{train})$

Algorithm 4 MR3 Training: Algorithm 4 the training procedure for a meta-regressor model designed for an elementary neural ensemble model.

Require: X_{train}, y_{train} ▷ Training data and labels
Ensure: $model_{best}$ ▷ Optimally tuned model

$MR \leftarrow MetaRegressor()$
$H \leftarrow \{num\_layers \in \eta_1, \ldots, \eta_n,$
  For each layer i **in** $1, \ldots, num\_layers$:
    $units\_i \in \delta_1, \ldots, \delta_n,$
    $activation\_i \in \zeta_1, \ldots, \zeta_n,$
    $l2\_reg \in \iota_1, \ldots, \iota_n,$
  $dropout \in \gamma_1, \ldots, \gamma_n$
  $learning\_rate \in \theta_1, \ldots, \theta_n, \}$
$MetaCV = MetaSearchCV(MR, H, cv)$
$MetaCV.fit(X_{train}, y_{train})$
$params_{best} = MetaCV.best_params$
$model_{best} = MetaRegressor(params_{best})$
$model_{best}.fit(X_{train}, y_{train})$

QFNU_CS at SemEval-2024 Task 3: A Hybrid Pre-trained Model based Approach for Multimodal Emotion-Cause Pair Extraction Task

Zining Wang, Yanchao Zhao, Guanghui Han, Yang Song
School of Computer Science, Qufu Normal University, Ri Zhao, China

Abstract

This article presents the solution of Qufu Normal University for the Multimodal Sentiment Cause Analysis competition in SemEval2024 Task 3.The competition aims to extract emotion-cause pairs from dialogues containing text, audio, and video modalities. To cope with this task, we employ a hybrid pre-train model based approach. Specifically, we first extract and fusion features from dialogues based on BERT, BiLSTM, openSMILE and C3D. Then, we adopt BiLSTM and Transformer to extract the candidate emotion-cause pairs. Finally, we design a filter to identify the correct emotion-cause pairs. The evaluation results show that, we achieve a weighted average F1 score of 0.1786 and an F1 score of 0.1882 on CodaLab.

1 Introduction

The competition of multimodal emotion cause analysis(Gandhi et al., 2023) involves not only understanding linguistic content but also recognizing and comprehending various forms of information such as emotional expressions, sounds, and images. The significance of this competition lies in its ability to comprehensively understand and interpret emotions and motivations in human communication. By analyzing various forms of information in conversations, we can more accurately identify the sources and reasons for emotions, thereby enhancing our understanding of human behavior and communication methods. This holds importance across various fields including psychology, human-computer interaction, and affective computing, aiding in the development of more intelligent and human-centric technologies and systems, improving communication efficiency and quality, and promoting better understanding and communication among individuals.

This paper details our contribution to SemEval-2024 Task 3: Multimodal Emotion Cause Analysis in Conversations(Wang et al., 2024), encompassing two sub-tasks: extracting emotion-cause pairs(Xia and Ding, 2019) from text-only dialogues and from multimodal dialogues that include text, audio, and video modalities. In this task, we place a particular emphasis on implementing Sub-task 2.

For Sub-task 2: Multimodal Emotion-Cause Pair Extraction, we aim to extract emotion-cause pairs from dialogues that contain representations in text, audio, and video. Each pair includes an emotional utterance, its emotion category, and a cause utterance. The challenge lies in integrating information from multiple modalities to accurately identify emotional expressions and their related causes.

In our approach to task 2, we first preprocessed the dataset, mapping the text, audio, and video data of the Emotion Cause in Friends (ECF) dataset to a unified feature space. Then, we utilized a baseline model with a two-stage training scheme: emotion recognition and cause pair extraction. This approach focused on utilizing modalities, selecting models such as Bert(Devlin et al., 2018) and LSTM(Yu et al., 2019), and adjusting parameters for two phases of model training. After that, we predicted on test data in two stages using the trained models and evaluated the results through CodaLab to obtain corresponding F1 scores.

Our best-performing solution involved using Bert for emotion recognition followed by LSTM for cause pair extraction across all three modalities, achieving an F1 score of 0.1882. This methodological progression demonstrates our systematic approach to tackling the complexities of multimodal emotion cause analysis, highlighting our efforts in dataset preprocessing, model experimentation, and performance evaluation, while also proving the effectiveness of the baseline model(Wang et al.).

2 Methodology

In this section, we describe the E-MECPE method

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 349–353

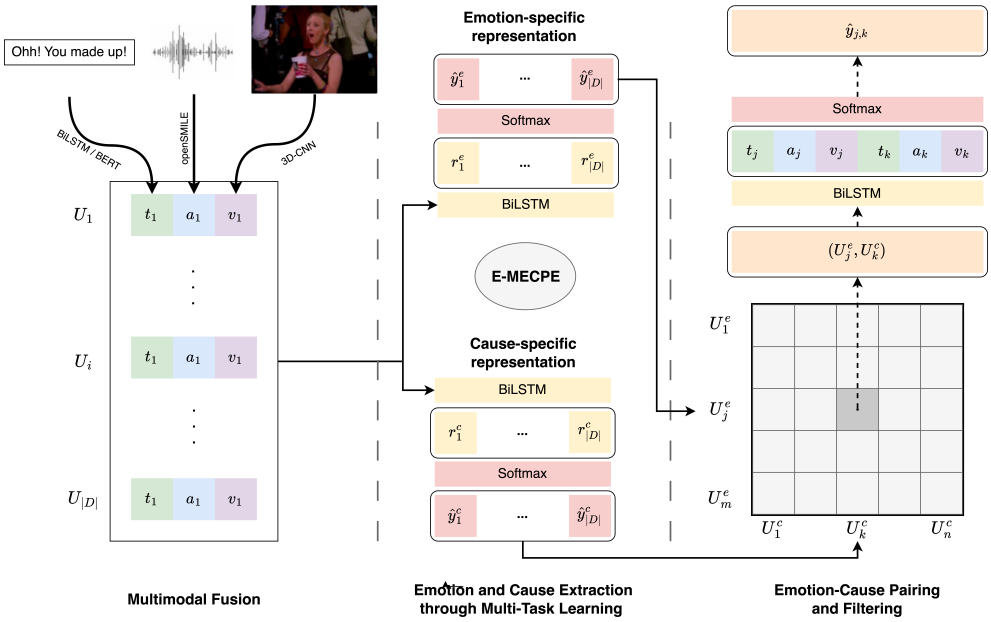


Figure 1: Framework diagram of the E-MECPE methodology

in depth. In general, this method is divided into three main parts: multimodal fusion, emotion and cause extraction through multi-task learning and emotion-cause pairing and filtering. The methodology of this paper is summarized in Fig. 1.

2.1 Multimodal Fusion

First, we obtain the representations of the three modalities from the text, audio and video modalities for their respective modalities. Then, the three modalities are stitched together in the order of text-audio-video to obtain the joint representation of the three modalities. The feature extraction method for each of these modalities is as follows:

For text, each token is initialized with pre-trained 300-dimensional GloVe vectors(Pennington et al., 2014). Subsequently, we used two different models to extract text features: the BiLSTM (Bidirectional Long Short-Term Memory Network) and the BERT (Bidirectional Encoder Representation Transformer).BiLSTM is a classical recurrent neural network that can effectively capture long-term dependencies in text sequences by a bidirectional structure that considers both forward and backward information. BERT, on the other hand, is a pre-trained language model based on the Transformer architecture, which is pre-trained on large-scale textual data and is able to capture rich semantic information. In this study, BiLSTM is used as a textual feature extractor to capitalize on its representational learning ability in sequential data; while BERT, as another textual feature extractor, acquires deeper semantic information by pre-training the model(Kim and Park, 2023). These two models are independently applied to discourse-level feature extraction tasks to evaluate their performance on sentiment and cause extraction tasks.

In the audio domain, we extract the 6373-dimensional acoustic features (a_i) using the openSMILE toolkit, leveraging the feature set designed for the INTERSPEECH 2013 Emotion Challenge. This comprehensive approach allows us to capture nuanced acoustic characteristics, providing a rich foundation for our subsequent analyses.

For video processing, we use a 3D-CNN network variant called C3D(Tran et al., 2015; Rao and Liu, 2020) to extract 16 frames from each video and process them through the C3D network to obtain 4096-dimensional video descriptors optimized for dimensionality reduction and to extract 128-dimensional visual features from each speech video.

2.2 Emotion Extraction

Our aim in sentiment extraction is to derive sentiment-related features from the discourse. We process each discourse to obtain sentiment-specific representations (re_i) by means of a specific bidirectional long short-term memory network (BiLSTM). The BiLSTM processes the forward and reverse sequences of the discourse separately by means of its two LSTM units, and ultimately combines the outputs of these two directions. This allows the network to take into account both the forward and backward information of the discourse, thus pro-

viding a more comprehensive understanding of the emotional content of the discourse.

Next, the sentiment-specific representation (re_i) is fed into a softmax layer, the output of which can be considered as the probability that the discourse belongs to each sentiment category. the formula for the softmax layer is as follows:

$$\hat{\mathbf{y}}_i^e = \text{softmax}\big(\mathbf{W}^e \mathbf{r}_i^e + \mathbf{b}^e\big) \tag{1}$$

where W_e and b_e are the weights and biases of the softmax layer, respectively, and $\hat{y}_i^e$ is the predicted sentiment distribution.

2.3 Cause Extraction

The purpose of the cause extraction part is to recognize causal relationships in discourse. We use another BiLSTM to extract cause-specific representations (rc_i). This BiLSTM works in a similar way to the BiLSTM used in sentiment extraction, but the parameters are not shared to ensure that the network learns the specific features for the cause extraction task.

The reason-specific representation (rc_i) is then fed into another softmax layer that focuses on determining the probability of different reason categories in the discourse. The formula for this softmax layer is as follows:

$$\hat{\mathbf{y}}_i^c = \text{softmax}\big(\mathbf{W}^c \mathbf{r}_i^c + \mathbf{b}^c\big) \tag{2}$$

Here, W_c and b_c are the weights and biases of this softmax layer, and $\hat{y}_i^c$ denotes the predicted cause distribution.

2.4 Loss calculation

Our goal is to minimize the cumulative loss of the model on the emotion extraction and cause extraction tasks. The total loss L_{total} is the sum of the losses of the two tasks and is calculated by the following formula:

$$\mathcal{L}_{\text{total}} = -\sum_{i=1}^{N} \left(\sum_{j=1}^{C} y_i^{e,j} \log(\hat{y}_i^{e,j}) + \sum_{k=1}^{K} y_i^{c,k} \log(\hat{y}_i^{c,k}) \right) \tag{3}$$

Where $y_{j,i}^e$ and $y_{k,i}^c$ denote the uniquely hot encoding of the true emotion and cause labels, respectively, N is the number of training samples, and C and K are the number of emotion and cause categories, respectively. This loss function ensures that the model learns to extract features related to emotions and reasons efficiently, thus improving the model's performance on both tasks.

2.5 Emotion-Cause Pairing and Filtering

Following the acquisition of Candidate Emotion Utterances and Candidate Cause Utterances, the pivotal task is to discern the existence of a causal relationship between sets E and C, ensuring the extraction of valid emotion-cause pairs. Initially, E and C are organized into a dot matrix, depicted in the third segment of Fig. 1, resulting in the generation of all conceivable candidate pairs denoted as $x(U_j^e; U_k^c)$. This vector amalgamates the self-contained multimodal representations of the emotion and cause expressions, along with a distance vector capturing the relational nuances between the two expressions.

The composite representation is then inputted into a softmax layer to determine the validity of the pairing x, filtering and extracting relevant emotion-cause pairs from numerous possibilities.

$$\hat{\mathbf{y}}_{j,k} = \text{softmax}\left(\mathsf{W}\mathbf{x}_{(\hat{U}_j^e, \hat{U}_k^e)} + \mathbf{b}\right) \tag{4}$$

3 Experiments

3.1 Data Resources

The official dataset consists of three modalities: text, audio, and video clips, and includes 1,374 conversations and 13,619 utterances annotated for 9,794 emotion-cause pairs across the three modalities. The relevant connections are stored in a JSON file and correspond to independent video segments through specified IDs. In order to fully utilize all the multimodal data, we first preprocess and reduce the dimensionality of the data according to the methods described in the paper.

Specifically, during the preprocessing stage, for the audio data in the video, we use the ffmpeg tool to extract the corresponding audio files for each video segment. We then utilize the open-source tool called openSMILE (Eyben et al., 2010) and apply The INTERSPEECH 2013 ComParE feature set (Schuller et al., 2013), which is the default feature set of openSMILE, to extract features from the audio data. As a result, we obtain a 6373-dimensional acoustic feature vector.For video data, we refer to the C3D model structure to extract video features and obtain a 4096-dimensional representation.As for text data, following the same approach as described in the paper, we utilize pre-trained Glove word vectors to obtain text embeddings.

3.2 Training

The training process is divided into two parts: the first part is emotion extraction and cause extraction,

and the second part is the extraction of emotion-cause pairs. We explored three different training conditions: utilizing only textual modalities, combining textual and audio modalities, combining textual and video modalities, and leveraging all data modalities and fine-tune the model parameters based on the baseline to select the appropriate parameters to obtain the best score.

Emotion extraction and cause extraction: The initial phase of our experiment compared emotion extraction using Bert and BiLSTM model architectures, conducted on an RTX 4070Ti Super GPU setup. Key training parameters were carefully selected to enhance model performance. The batch size for BiLSTM was fixed at 16, while for Bert, it was set to 4, with the training spanning 15 epochs. The loss weights for both emotion extraction and cause extraction tasks were set to 1.0, indicating their equal importance in our training objectives.

Emotion-cause pairs extraction: In the subsequent phase focusing on cause pair identification, the same model architecture was employed, trained under identical conditions to assess the effect of data modality on performance. The batch size was increased to 200 to potentially improve generalization, with a learning rate of 0.005 aimed at optimal convergence. A 0.5 dropout keep probability for word embeddings was introduced for added regularization, while maintaining a 1.0 keep probability for the softmax layer. The l2 regularization coefficient remained at 1e-5, consistent with our approach to model complexity control.

3.3 Evaluation

Similar to baseline, we utilize the macro-averaged F1 score (Gui et al., 2018) as the primary evaluation metric for our task. This metric accounts for both precision and recall, providing a balanced assessment of model performance. The F1 score is calculated using the following formula:

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{5}$$

where:

- P denotes precision, calculated as the ratio of correctly predicted emotion-cause pairs to the total predicted pairs.
- R denotes recall, calculated as the ratio of correctly predicted emotion-cause pairs to the total annotated pairs.

In our evaluation, F_1 is the harmonic mean of precision and recall, indicating the model's balance in detecting emotion-cause pairs: a higher F_1 score signifies better performance.

Table 1: Experimental Results

Model	Modality	$F1_{emotion}$	$F1_{caution}$	$F1_{pair}$
BiLSTM	T	0.7441	0.7008	0.5041
	TA	0.7398	0.6986	0.5104
	TV	0.7431	0.7016	0.5162
	TAV	0.7422	0.6993	0.5226
Bert	T	0.7362	0.6687	0.5104
	TA	0.7356	0.6637	0.5160
	TV	0.7365	0.6700	0.5104
	TAV	0.7363	0.6648	0.5246

3.4 Results and analysis

We assessed Bert and BiLSTM models on various modalities: text (T), text-audio (TA), text-video (TV), and their combination (TAV), as shown in Table 1. Results underline the models' proficiency in extracting sentiment-cause pairs from multimodal dialogues, with distinct performance variations across modalities.

The BiLSTM model demonstrates incremental improvements in $F1_{pair}$ scores from T to TAV, indicating the advantage of utilizing multimodal data. The highest performance is observed in the TAV setup with a score of 0.5226, underscoring the benefits of combining text, audio, and video.

Conversely, the Bert model showcases superior performance in the TAV modality, achieving an $F1_{pair}$ score of 0.5246. This performance highlights Bert's ability to effectively leverage deep contextual embeddings across modalities for more accurate extraction of sentiment-cause pairs. The robustness of Bert, particularly in the multimodal TAV setup, confirms its efficacy in handling complex multimodal data.

Overall, Bert emerges as the preferred model for extracting sentiment-cause pairs across all modalities, with a peak performance in the TAV configuration, reflected by a weighted average F1 score of 0.1786 and an F1 score of 0.1882 on CodaLab. These findings advocate for the continued exploration of multimodal approaches, particularly leveraging models like Bert that excel in contextual understanding and integration of multimodal data.

4 Conclusion

In this paper, we present Effective Multimodal Emotion-Cause Pair Extraction (E-MECPE)

method. We used this method to perform emotional cause analysis on the Emotion-Cause-in-Friends (ECF) dataset. Ablation experiments were conducted for text unimodal and multimodal under different text encoders, respectively, and the relevant parameters associated with the experiments were tuned. The experimental results show that BERT encoding-based text representation and multimodal joint representation help in the extraction of emotional cause pairs, and that the parameter settings are crucial for the performance enhancement of this task. This finding not only validates the effectiveness of our method, but also points out an important direction for future research in the field of sentiment analysis kresearch by pointing out an important direction.

5 Prospects for Advancement

Due to the late entry time, limited hardware resources, and short submission period, we only had time to fine-tune and conduct ablation experiments based on the baseline. However, we believe that there is still a lot of room for improvement in adjusting this model. For example, further attempts can be made in aligning and filtering methods for multimodal data, selecting more encoders, and enhancing the model's understanding of causal relationships. We will also continue exploring on top of this model to continuously advance the development of this research direction.

6 Acknowledgements

This research is supported by Natural Science Foundation of Shandong Province [grant number ZR2022QF088]; Natural Science Foundation of Rizhao [grant number RZ2021ZR30]; and ShanDong Students' Platform for innovation and entrepreneurship training program [grant number S2023104460124]

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1459–1462, New York, NY, USA. Association for Computing Machinery.

Ankita Gandhi, Kinjal Adhvaryu, Soujanya Poria, Erik Cambria, and Amir Hussain. 2023. Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions. *Information Fusion*, 91:424–444.

Lin Gui, Ruifeng Xu, Dongyin Wu, Qin Lu, and Yu Zhou. 2018. Event-driven emotion cause extraction with corpus construction. In *Social Media Content Analysis: Natural Language Processing and Beyond*, pages 145–160. World Scientific.

Kyeonghun Kim and Sanghyun Park. 2023. Aobert: All-modalities-in-one bert for multimodal sentiment analysis. *Information Fusion*, 92:37–45.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Chengping Rao and Yang Liu. 2020. Three-dimensional convolutional neural network (3d-cnn) for heterogeneous material homogenization. *Computational Materials Science*, 184:109850.

Björn Schuller, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, Erik Marchi, et al. 2013. The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. In *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France*.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270.

NewbieML at SemEval-2024 Task 8: Ensemble Approach for Multidomain Machine-Generated Text Detection

Bao Tran and **Nhi Tran**
Ho Chi Minh City University of Technology, VNU-HCM, Ho Chi Minh City, Vietnam
{bao.tran2003, nhi.tranluongyen}@hcmut.edu.vn

Abstract

Large Language Models (LLMs) are becoming popular and easily accessible, leading to a large growth of machine-generated content over various channels. Along with this popularity, the potential misuse is also a challenge for us. In this paper, we use subtask A monolingual dataset with comparative study between some machine learning model with feature extraction and develop an ensemble method for our system. Our system achieved 84.31% accuracy score in the test set, ranked 36th of 137 participants. Our code is available at: https://github.com/baoivy/SemEval-Task8

1 Introduction

In our current time, large language models (LLMs), such as ChatGPT (Ouyang et al., 2022), GPT-4[1], LLaMA (Touvron et al., 2023) and BLOOMz (Muennighoff et al., 2023) can be easily observed to be becoming increasingly prevalent from diverse forms ranging of news, multimedia to education. How outstandingly LLMs answer to user's problems makes them appealing for automatic missions as well as diminishing human labor in many scenarios. Nevertheless, this also unexpectedly leads to problems with regard to human's misuses, spreading misinformation and causing disruptions in the education system in particular. Therefore, it is necessary to develop systems that can automatically distinguish AI contexts from human-written ones.

Recently, with the exponential growth of LLMs, many researchers have attempted to distinguish human-written texts from machine- generated ones. Uchendu et al., 2021, Wang et al., 2024b, He et al., 2024, Liu et al., 2023b have shown us about machine-generated and human writing data from various source. Mitchell et al., 2023, Bao et al., 2023, Deng et al., 2023 used zero-shot classification method to calculate probabilities from perturbated input text. Bhattacharjee and Liu, 2023 leveraged prompt-base method to utilize LLMs as detector. Gehrmann et al., 2019 used statistical method to detect machine-generated paragraph with language model to compute conditional probability. For fine-tuning language model method, Fagni et al., 2021 had a comparative study among pre-trained language model, feature-base and character level classification on DeepFake dataset and showed that pre-trained language model has a best result than others. Liu et al., 2023c used feature-base classification with RoBERTa as embedding and LSTM + Self-Attention in classification head. Bhattacharjee et al., 2023 used unsupervised and self-supervised learning by leveraging domain adaptation on unlabeled dataset and contrastive learning belonging with pre-trained language model to learn domain representations. Kirchenbauer et al., 2023, Liu et al., 2023a used a novel approach with watermark embedding to detect LLMs text that employed by LLMs or neural network.

Being inspired by feature-based classification technique, we propose to have a comparative study for simple and lightweight machine learning method beside the trend of LLM. Our system compares various machine learning models among with ensemble method for multiple machine learning method to find the best combination for our system.

The rest of the paper is as follows. The section 2 generalizes task description and dataset for our experiment. The section 3 shows the description of our system. The experimental setup and results are presented in section 4. Finally, section 5 is the conclusion and discussion about our work.

2 Task description & dataset

2.1 Task description

In SemEval 2024- Task 8 (Wang et al., 2024a), the topic for subtask A is ***Binary Human-Written vs.***

[1] https://openai.com/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 354–360
June 20-21, 2024

Machine-Generated Text Classification. The full text is to determine whether an essay is human-written or machine-generated. There are two tracks for subtask A: monolingual (only English sources) and multilingual. On this subtask, we only focus on monolingual dataset.

2.2 Dataset overview

The SubTask-A monolingual dataset originated from various sources of content, including Wikipedia, WikiHow, Reddit, arVix, PeerRead, and OutFox (Koike et al., 2023). According to the author, this is an extended version of the M4 dataset (Wang et al., 2024b). All paragraphs in the dataset of subtask A monolingual are written in English. This dataset contains a total of 159,029 essays, which were split into a three-part train set, development set (dev set), and test set. The monolingual dataset contains two types of labels, 0 represented by human writing and 1 represented by machine-generated. In particular, the train set was constructed from 5 different generator (Human, ChatGPT, Dolly-v2, CoHere and Davinci003) and the development set was constructed only from Bloomz. For the test set, GPT-4 had been added along with the remaining generator to generate essays. The overview statistical and distribution of labels will be detailed in Table 1 and an example of the dataset is represented in Table 2. Additionally, the distribution between human labels and machine-generated labels on the train set is almost equal so the class imbalance technique is not used for this task.

Moreover, a pre-processing step was applied to the dataset by the following deletions and changes:

- Removing punctuation in sentence
- Lower casing text
- Removing any leading and trailing whitespace
- Remove URLs

Dataset	***#Number***	**Label Distribution (%)**	
		Human	Machine
Train set	119,757	52.9	47.1
Dev set	5,000	50.0	50.0
Test set	34,272	47.5	52.5

Table 1: Dataset statistical

3 System Description

We will describe our developed model in this section. On Section 3.1, we will discuss how we embedded sentences in essays using a pre-trained language model. Then, for the crucial section, we would like to present the detail of our model on 3.2. We perform our architecture based on Figure 1. First step, the essays need to be embedded through pre-trained language model. Next, we use ensemble method with base model and then give a final prediction by using meta-model.

3.1 Embedding

For the embedding stage, each token needs to be represented in a vector. Some essays have more than 512 tokens, which will lead to exceed at original BERT (Devlin et al., 2019), we determine to utilize Longformer (Beltagy et al., 2020) model to capture semantic embedding of each word within essays of dataset. Given the essay X, the vector embedding of each token will be calculated in the essay. The input will be formed as (w_i is a word in essay):

$$\texttt{<s>}\ w_1\ w_2\ \ldots\ w_i\ \texttt{</s>}$$

This produces an embedding matrix $\mathbf{E} \in \mathbb{R}^{N \times K}$ (K is a hidden size of word, N is a number of token) by taking the last hidden layer. After that, mean pooling is applied to each vector embedding of the matrix to flatten into a standard vector to aggregate feature of token. The dimension of the vector for each essay will be $\mathbf{X} \in \mathbb{R}^N$ where N is a number of tokens in essay and $\mathbf{X}$ is a feature vector of essay X.

3.2 Ensemble model

After text embedding, we develop our classification stage for the system. We will discuss each base model in section 3.2.1 and how we ensemble various base models in section 3.2.2

3.2.1 Base model

We utilized Support Vector Machine (SVM), XGBoost, Logistic Regression, and K-nearest neighbors (KNN) as the base model for our ensemble method.

Support Vector Machine (SVM) (Hearst et al., 1998) is a supervised learning model that is used for classification and regression tasks. SVM maximizes the hyperplane or set of hyperplanes to find the best boundary that separates different classes in a dataset.

ID	Essays	Label
1	...Step 10. Pause The Game. To pause the game, just press the "start" button...	Machine
56406	...If you haven't used it in the last six months there is little chance you'll use it in the next six months. Toss it.	Human

Table 2: Example dataset

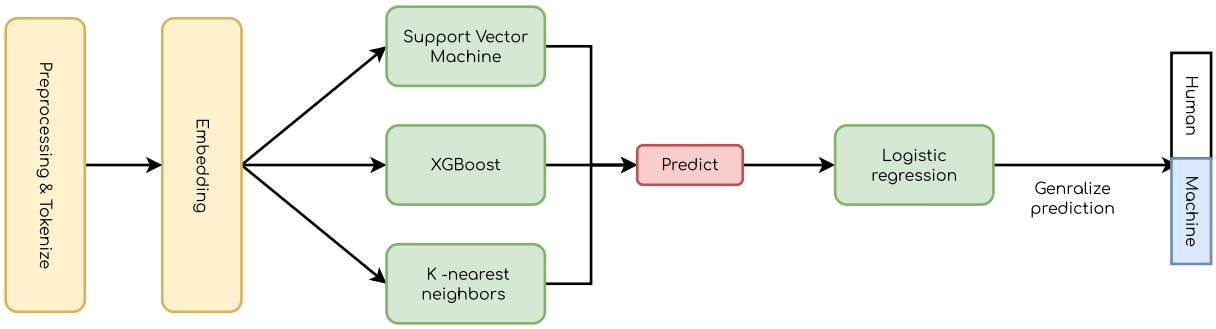


Figure 1: Overview of our system architecture. We demonstrate the best combination model from our experiment

Hyperparameter	Value
C	1.0, 10.0
λ	0.0001, 0.001, 0.1, 0.2, 0.5, 1, 10

Table 3: SVM configuration

Hyperparameter	Values
Learning rate	0.1, 0.2
Estimators	60, 80, 100
Max depth	2, 4, 6

Table 4: XGBoost configuration

Due to the non-linearity of the dataset, we decided to use the Radial basis function (rbf) kernel for SVM, which is defined as Formula 1. Moreover, to find the best parameter for the SVM model, we listed all the hyperparameter values of C and λ used in the grid search as Table 3.

$$K = Ce^{-\gamma||x-z||^2} \quad (1)$$

XGBoost (Chen and Guestrin, 2016) is a scalable end-to-end tree boosting technique which allows to correct the error of the previous tree by creating multiple trees sequentially. The classifier also assigned a weight value to each independent variable and used some techniques to prevent overfit like tree pruning, sparsity awareness, etc.

Same as SVM, we construct candidate hyperparameter values of max depth of the tree and λ used in the grid search as Table 4.

Logistic Regression (LR) is a simple technique for binary classification. Given feature variables, the output is a probability from $[0; 1]$. This can be achieved by applying of a sigmoid function to the linear combination of the independent variables. In our system, we only use the default configuration.

K-nearest neighbour (KNN) is a simple technique for classification which uses majority vote on k closest data point to target point. Grid search is also utilized to find the best value of k, where $k \in \{3, 5, 7, 9\}$.

3.2.2 Stack ensemble

Ensemble different machine learning models is the way to improve prediction accuracy to leverage the strengths and mitigate the weaknesses of the individual base models. In our system, we choose stack ensemble as an ensemble method for our system. For this technique, applicable in scenarios with N base model $(M_1, M_2, ..., M_n)$ and meta-model M, determine meta feature for meta model by predicting each base model $\hat{\mathbf{X}} = (M_1(X), ..., M_n(X))$. Then predict the final output by calculating meta model $y = M(\hat{\mathbf{X}})$. Many different base models and meta model have been evaluated and compared, including Naive Bayes, k-nearest neighbors, SVM, XGBoost, and Logistic Regression as section 3.2.1

3.3 Experiment setup

We describe our system setup procedure. As GPU, we use a single RTX 4090 24GB to train and infer our system for both stages. In the embedding stage, we use Hugging Face[2] library for the Longformer model. For maximum token length in the pre-trained language model, because some essays are longer than 512 tokens, we set it to a maximum of 1024 tokens with padding and truncating. We infer each essay without any training on it.

For the classification stage, with SVM, Logistic Regression, and KNN model, due to the large dimension of the dataset, we proposed to use cuML[3] library, which supports GPU-accelerated for machine learning algorithms. For XGBoost we use xgboost [4] library and sklearn[5] for stack ensemble. Hyperparameter tuning is used in each machine learning model to find the best parameter for each model (all candidate parameters are defined in section 3.2.1). A 5-fold cross-validation is used to find the optimal configuration for the ensemble.

4 Results & Discussion

4.1 Evaluation metric

For subtask A monolingual task, the metrics used to evaluate our result for the dataset are Marco-F1, Mirco-F1, and Accuracy. The main metric for ranking submission is Accuracy. In more detail, the accuracy metric is given by the ratio of the total number of correct predictions to the total predictions done by the model, regardless of true or false predictions. Micro F1-score is the harmonic mean of precision and recall and macro-F1 score is defined as the average of Mirco F1-score across different classes.

4.2 Results

In this section, we present the result of our model, focusing on its accuracy in the dev set and test set. Table 5 shows the performance of our models with some combination with the base model when using the ensemble method mentioned in section 3.2, compared to the dev set. All results were running on the best hyperparameter value of each base model. We first compare the efficiency of each individual model. SVM gives the best performance among all (0.6986). Surprisingly, LR have better performance than XGBoost in monolingual task.

From the result of each model, we also compare our main model with some combinations of the base model when using the stack ensemble method which is represented in Table 5. SVM is used as base models for all ensemble experiments since they give better performance than others. The results do not significantly differ from the three models in our experiment. For model 3, the result is slightly better than model 1 and model 2 which achieved 0.7101 accuracy score.

For the test set, we can notice that all results from 7 methods are not significant differences. The result of SVM (0.8399) still outperformed on individual tests. However, XGBoost has surpassed the performance of KNN and LR on the test set (0.8319 compared to 0.8244 and 0.8155). The LR has the worst performance among 4 models. Surprisingly, after evaluating the test set on the ensemble method, model 1 inferior when compared to the rest. In contrast, model 3 has the best result at the test set with an accuracy of 0.8458. We also visualize our performance of model by representing the confusion matrix in Figure 2

Table 6 shows the result at the stage. We evaluate our result on model 2. Our system achieves 0.8438 which is ranked 36th out of 137. Unfortunately, we can not surpass the result of baseline (achieves 0.8847), which is using RoBERTa model (Zhuang et al., 2021) for classification. Besides, this is a prospective result that can achieve to acceptable score when comparing the traditional machine learning method with the pre-train language model and LLMs. Moreover, we can have an insight into training on traditional machine learning methods and language models nowadays. We believe that if we have a better strategy on hyperparameter tuning, the result could be higher than our official submission.

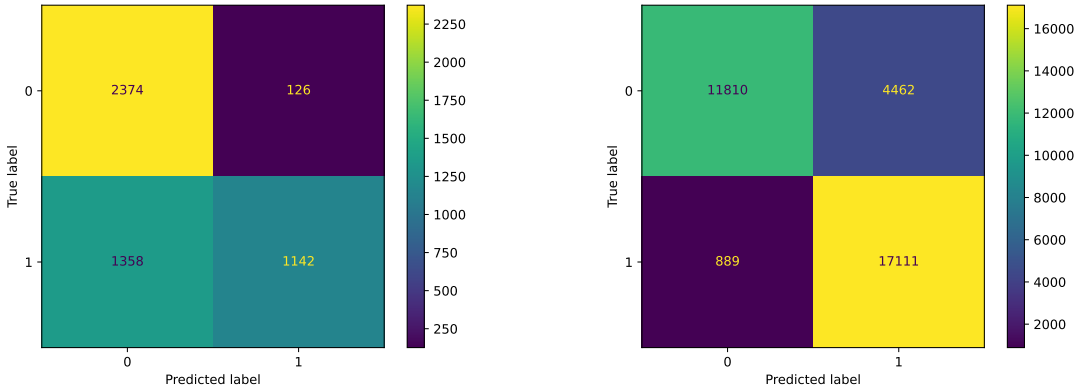
5 Conclusion

In subtask A monolingual of SemEval task 8, we have represented our system for machine-generated detection. We proposed to develop our system based on ensemble of multiple traditional machine learning method with hyperparameter tuning. We found that XGBoost, SVM and KNN as base model and Logistic Regression in meta model would give the highest result. Our official system was ranked the 36th to 137 in test set of subtask A monolingual

[2] https://huggingface.co/
[3] https://github.com/rapidsai/cuml
[4] https://github.com/dmlc/xgboost
[5] https://scikit-learn.org/

Method	Development phase			Test phase		
	Accuracy	Micro F1	Macro F1	Accuracy	Micro F1	Macro F1
SVM	**0.6986**	**0.6986**	**0.6758**	***0.8399***	***0.8399***	***0.8360***
XGBoost	0.6486	0.6486	0.6206	*0.8319*	*0.8319*	*0.8293*
KNN	0.5492	0.5392	0.5057	*0.8244*	*0.8244*	*0.8228*
LR	0.6774	0.6774	0.6549	*0.8155*	*0.8155*	*0.8114*
Model 1	**0.7108**	**0.7108**	**0.6925**	*0.8329*	*0.8329*	*0.8279*
Model 2	0.7032	0.7032	0.6840	0.8439	0.8439	0.8401
Model 3	0.7028	0.7028	0.6832	***0.8458***	***0.8458***	***0.8422***

Table 5: Result on different method on dev set and test set. Test result with italicized have been run after test phase deadline. Denoted that Model 1 is XGBoost + SVM as base model, LR as meta model, Model 2 is XGBoost + SVM + KNN as base model, LR as meta model, Model 3 is LR + SVM + KNN as base model, XGBoost as meta model.



(a) (b)

Figure 2: Confusion matrices for (a) the development set and (b) the test set on official submission

Team	Subtask A - Monolingual Accuracy
Baseline	0.8847
#1 Team	**0.9688**
Ours (36th)	0.8438

Table 6: Result and ranking on test set

with 0.8439 accuracy score and 0.7032 accuracy score in dev set. From our result, traditional machine learning methods still have been proven effective in classification, with some training strategy, compared to other methods such as LLMs.

Acknowledgements

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this research.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Amrita Bhattacharjee and Huan Liu. 2023. Fighting fire with fire: Can chatgpt detect ai-generated text?

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, KDD '16. ACM.

Zhijie Deng, Hongcheng Gao, Yibo Miao, and Hao Zhang. 2023. Efficient detection of llm-generated texts with a bayesian surrogate model.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. Gltr: Statistical detection and visualization of generated text.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples.

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S. Yu. 2023a. An unforgeable publicly verifiable watermark for large language models.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023b. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023c. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji,

Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Hidetsune at SemEval-2024 Task 3: A Simple Textual Approach to Emotion Classification and Emotion Cause Analysis in Conversations Using Machine Learning and Next Sentence Prediction

Hidetsune Takahashi
Waseda University
takahashi78h@toki.waseda.jp

Abstract

In this system paper for SemEval-2024 Task 3 subtask 2, I present my simple textual approach to emotion classification and emotion cause analysis in conversations using machine learning and next sentence prediction. I train a SpaCy model for emotion classification and use next sentence prediction with BERT for emotion cause analysis. While speaker names and audio-visual clips are given in addition to text of the conversations, my approach uses textual data only to test my methodology to combine machine learning with next sentence prediction. This paper reveals both strengths and weaknesses of my trial, suggesting a direction of future studies to improve my introductory solution.

1 Introduction

SemEval 2024 Task 3 (Wang et al., 2024) calls for assigning an emotion to each utterance and extracting its emotion cause in conversations. Subtask 2, which I participate in, requires emotion classification and identification of emotion cause utterances with audio-visual clips available whereas subtask 1 requires identification of specific textual span as well without audio-visual clips.

I participate in subtask 2, for which a speaker name, text and an audio-visual clip are given for each utterance. Instead of not identifying specific cause span in the emotion cause utterance in this subtask, I set a limitation to use textual data only while audio-visual data are also available. Therefore, my methodology uses textual data of utterances only as input to classify emotions and identify cause utterance numbers as output. For this reason, training data from subtask 1, for which video names are not given, are used instead of data from the subtask I participate in.

While the task (Wang et al., 2024) prohibits use of additional annotation data, I overlooked the sentence stating the rule and mistakenly used additional data for my solution. I would like to show my appreciation for the task organizers and readers acknowledging and understanding my mistake of using additional data.

For data preparation, official training data for subtask 1 (CSV converted version) (Wang et al., 2023), training data (translated and CSV converted version) from SemEval2024 Task10 subtask 1 (ERC) by Kumar et al. (2023) and data by Nikam (n.d.) are used. They all are concatenated in that order and adjusted so that the resulting dataset has first 7001 neutral utterances (including 7000th counting from 0) and first 5001 utterances at maximum for each emotion other than neutral.

Then, SpaCy-v3 model (Kömeçoğlu, 2023) is trained using the adjusted training data for emotion classification. In addition to that, next sentence prediction (Cathrine, 2023) is used for identification of emotion cause utterances. In that step, I decided to simplify the methodology by hypothesising that the emotion cause utterance is the utterance itself or its previous utterance. With this simple assumption, my algorithm checks the relatedness of each utterance and its previous utterance using next sentence prediction (Cathrine, 2023), which returns true or false. Previous utterance is chosen as cause utterance if the two utterances are deemed related, and the utterance itself if not.

The result shows a limited performance of my introductory solution, but it also clarifies a direction to its improvement. Although my combined methodology has a large room for improvement, it does have a potential in its simplicity and limitation to use textual data only. This paper aims to share an experimental trial to test my combined methodology, guiding a direction to its future application and improvement.

My code is available on GitHub [1].

[1] https://github.com/Hidetsune/SemEval2024_Task3.git

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 361–364

2 Background

The subtask I participate in (subtask 2) focuses on emotion classification and emotion cause analysis with text data and audio-visual clips. Subtask 1, on the other hand, does not allow participants to use audio-visual clips and requires extracting specific textual cause spans as well. My participation in subtask 2 sets a limitation to use textual data only, which means that it is substantially the same as subtask 1 except that I do not extract specific textual cause spans. Training data from subtask 1 are used instead of that from subtask 2 because they seem to be identical to each other except that they have no video names in the dataset. Therefore, my methodology for subtask 2 uses data from subtask 1 and additional data from other sources for training. Given the evaluation dataset with audio-visual clips available, my methodology, which is trained by textual data only, assigns an emotion category and its cause utterance as output using textual data of the evaluation dataset.

This task is technically a mixture of two topics, which are emotion classification and emotion cause analysis. As for emotion classification, many previous studies have been conducted especially on social media including Twitter and Facebook. For instance, a work by Gaind et al. (2019) classifies text on social media into six emotion categories with high accuracy. Another study by Bryniels-son et al. (2014) investigates in people's emotions during crises using a support vector machine. In addition to its use for social media, its application to real conversations is also getting an attention. A study by Graterol et al. (2021), for example, applies emotion detection to social robotics, aiming to improve its ability to interpret feelings of humans from a viewpoint of NLP methods.

There are many previous studies for emotion cause analysis too. A study by Fan et al. (2019), for example, uses hierarchical neural network to get high accuracy. Another study by Ding et al. (2020) adopts a complicated approach, resulting in reliable accuracy.

On the other hand, this paper aims to test a simple approach to combine classical machine learning method with next sentence prediction with a certain assumption. My methodology has a strength in its simplicity, but the result shows a large room for improvement.

3 System overview

The main strategy of my system is a combination of classical machine learning method with next sentence prediction. Machine leaning is used for emotion classification and next sentence prediction is used for identification of emotion cause utterances. Audio-visual clips are available in this subtask, but only textual data of the utterances are used for my solution. A quick overview of my combined methodology is as follows.

1. **Training data preparation**: Official training data from subtask 1 (Wang et al., 2023) are converted from a json file into a pandas dataframe. Similarly, training data from SemEval2024 Task10 subtask 1 (ERC) (Kumar et al., 2023) are translated into English and converted into a pandas dataframe. The converted dataframes and data by Nikam (n.d.) are concatenated to compose an adjusted training data. The adjusted data have two columns, in which text and an emotion are stored respectively for each utterance.

2. **Emotion classification using machine learning**: Using the adjusted training data, SpaCy-v3 model (Kömeçoğlu, 2023) is trained and used for emotion classification. It assigns an emotion to each utterance from "neutral", "surprise", "anger", "sadness", "joy", "disgust" and "fear".

3. **Emotion cause utterance identification using next sentence prediction**: If an assigned emotion is not "neutral", next sentence prediction (Cathrine, 2023) identifies its emotion cause utterance. My methodology works under the simple assumption that emotion cause utterance is the utterance itself or its previous utterance.

In the first step, training data are prepared from multiple sources. The official training data from subtask 1 (Wang et al., 2023) are imported as a json file and converted into a pandas dataframe with text and an emotion for each utterance. Here, data from subtask 1 are used instead of that from subtask 2 because it is likely that the data are identical to each other except that video names are not given for data of subtask 1. Then, the resulting pandas dataframe, translated and converted version of training data from SemEval2024 Task10 (Kumar

et al., 2023) and data by Nikam (n.d.) are imported via CSV file format. As for the two additional datasets, irrelevant columns are dropped so that they are composed of two columns, in which text and emotions are stored respectively. They are concatenated into one dataframe and adjusted to have 7001 utterances (including 7000th counting from 0) for neutral and 5001 at maximum for each one of the other emotions related to this task (surprise, anger, sadness, joy, disgust and fear).

After this process of data preparation, SpaCy-v3 model (Kömeçoğlu, 2023) is trained using the prepared training data. An unlabeled evaluation dataset is imported as a json file, and the trained model assigns an emotion to each utterance.

At the same step, next sentence prediction with BERT (Cathrine, 2023) assigns an utterance number of emotion cause to each utterance if its assigned emotion is not "neutral". As stated before, it is hypothesised that the emotion cause utterance is either the utterance itself or its previous utterance. Under this assumption, next sentence prediction (Cathrine, 2023) checks whether or not an utterance that it is looking at is related to its previous utterance. The previous utterance is chosen as its emotion cause utterance if these are deemed related, and the utterance itself is chosen if not. After all these processes, lists that include emotions with the utterance numbers and emotion cause utterance numbers (['2_sadness', '1'] for example) are added to the original evaluation data for submission.

My participation in this task using the combined methodology reveals its limitations of the simple approach to emotion cause analysis. Since my methodology trains SpaCy-v3 model with over 33000 utterances, it is more natural to assume that the simplistic application of next sentence prediction is the main reason for the limited accuracy. My algorithm takes only the utterance itself and its previous utterance into account as possible emotion cause utterances. This premise does not allow my solution to cover cases where one utterance has an influence beyond multiple utterances, limiting the ability to deal with the entire conversation from a macroscopic point of view. On the other hand, there is no doubt that the accuracy of emotion classification is also a reason for the limited ability of my trial. Emotion cause utterances are assigned to non-neutral emotion utterances only, meaning that the algorithm loses its accuracy for both emotion classification and emotion cause analysis at a time if the trained model mistakenly assigns "neutral" to non-neutral emotion utterances.

4 Experimental setup

For the emotion classification part, multiple datasets needed to be processed to make an adjusted training dataset.

First of all, the official training data for subtask 1 (Wang et al., 2023) are imported as a json file. In the dataset, a conversation ID is assigned to each conversation, and one conversation has multiple utterances. For each utterance, an utterance ID, text, its speaker name and an emotion are assigned. The json file is converted into a pandas dataframe to make the data easier to deal with. Conversation IDs, utterance IDs and speaker names are dropped from the dataframe so that it has only "text" and "emotion" as columns. Data from SemEval2024 Task10 (Kumar et al., 2023), which have Hindi-English code-mixed utterances, are translated into English and converted into a pandas dataframe similarly. Data by Nikam (n.d.) are also imported as a pandas dataframe, and irrelevant columns of the two additional datasets are dropped so that they have text utterances and emotions as columns only. The number of utterances for each different emotion category is as shown on Table1. After these processes, the three dataframes (official training data for subtask 1, data from SemEval2024 Task10 (Kumar et al., 2023) and data by Nikam (n.d.)) are concatenated into one dataframe in that order. Only first 7001 (including 7000th counting from 0) neutral emotion utterances and first 5001 utterances for each one of the other emotions are extracted, dropping all the utterances that exceed the limitation from the concatenated dataset to compose an adjusted training data.

After this data preparation step, SpaCy-v3 model (Kömeçoğlu, 2023) is trained using the adjusted training data, and the trained model is used for emotion classification of the unlabeled evaluation dataset. Next sentence prediction (Cathrine, 2023) is also used for emotion cause analysis as stated in the previous section.

5 Results

In the evaluation phase, my solution was tested using the unlabeled test dataset. The result shows a limited ability of my approach, which combines classical machine learning with next sentence prediction under a simplistic assumption. The scores

Dataset	Anger	Disgust	Fear	Joy	Neutral	Sadness	Shame	Surprise	Contempt
Data from subtask1	1615	414	373	2301	5929	1147	0	1840	0
Data from Task10	819	127	514	1596	3909	558	0	441	542
Data by Nikam	4286	856	5409	11037	1811	6719	146	4062	0

Table 1: Datasets and emotion categories

w-avg. F1	F1	Ranking
0.1288	0.1389	12/16

Table 2: Task scores in evaluation phase

are displayed in Table 2.

6 Conclusions

To summarize, my methodology sets a limitation to use textual data only, testing a simple algorithm with a certain premise. I use classical machine learning for emotion classification, and next sentence prediction for identification of emotion cause utterances.

The next sentence prediction (Cathrine, 2023) in my simple approach takes only the utterance itself and its previous utterance into account, limiting its ability to cover the entire conversation from a macroscopic viewpoint. In addition to that, the accuracy of emotion classification between neutral and non-neutral turned out to be more important than previously thought since it has a significant effect on identification of emotion cause utterances as well.

Although the trial of my simple approach has a large room for improvement, it clearly guides a direction to its future studies. With improvements to enhance the ability to cover conversations from a macroscopic point of view, it might open the door for the potential of my combined methodology.

References

Joel Brynielsson, Fredrik Johansson, Carl Jonsson, and Anders Westling. 2014. Emotion classification of social media posts for estimating people's reactions to communicated alert messages during crises. *Security Informatics*, 3:1–11.

Jeeva Cathrine. 2023. Next sentence prediction with BERT. *scaler*.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 3574–3583.

Chuang Fan, Hongyu Yan, Jiachen Du, Lin Gui, Lidong Bing, Min Yang, Ruifeng Xu, and Ruibin Mao. 2019. A knowledge regularized hierarchical approach for emotion cause analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5618–5628. Association for Computational Linguistics.

Bharat Gaind, Varun Syal, and Sneha Padgalwar. 2019. Emotion detection and analysis on social media. *arXiv preprint arXiv:1901.08458*.

Wilfredo Graterol, Jose Diaz-Amado, Yudith Cardinale, Irvin Dongo, Edmundo Lopes-Silva, and Cleia Santos-Libarino. 2021. Emotion Detection for Social Robots Based on NLP Transformers and an Emotion Ontology. *Sensors*, 21(4).

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Başak Kömeçoğlu, Buluz. 2023. Emotion Classification with SpaCy v3 Comet.

Sanket Nikam. n.d. Emotion Detection in Text using Natural Language Processing.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

CLTeam1 at SemEval-2024 Task 10: Large Language Model based ensemble for Emotion Detection in Hinglish

Ankit Vaidya* , **Aditya Gokhale*** , **Arnav Desai*** , **Ishaan Shukla*** , **Sheetal Sonawane**
Pune Institute of Computer Technology
{ankitvaidya1905, adityangokhale, arnavdesai235, ishaanshukla10}@gmail.com,
sssonawane@pict.edu

Abstract

This paper outlines our approach for the ERC subtask of the SemEval 2024 EdiREF Shared Task. In this sub-task, an emotion had to be assigned to an utterance which was a part of a code-mixed dialogue. The utterance had to be classified into one of the following classes - disgust, contempt, anger, neutral, joy, sadness, fear, surprise. Our proposed system makes use of an ensemble of language specific RoBERTA and BERT models to tackle the problem. A weighted F1-score of 44% was achieved by our system. We conducted comprehensive ablations and suggested directions of future work. Our codebase is available publicly[1].

1 Introduction

Language has been the primary mode of communication for humans since pre-historic times. In linguistics, code-mixing traditionally refers to the embedding of words or phrases into an utterance of another language (Myers-Scotton, 1993). In many multi-lingual societies we see the development of code-mixed languages. Hinglish is one such language which is a linguistic blend of Hindi and English which is spoken primarily in India. Hinglish generally refers to Hindi that is written in the roman script and is used in combination with some English phrases. The variance in spellings and the multiple interpretations of Hindi words, depending on specific contexts, pose challenges for the analysis of language.

The SemEval workshop (co-located with NAACL 2024) explores and advances the current state of semantic analysis to tackle increasingly complex problems in natural language semanitcs. This paper outlines our approach for the Emotion Recognition in Coversation (ERC) (Kumar et al., 2023) sub-task of the Emotion Discovery and Reasoning its Flip in Conversation (EdiREF) (Kumar et al., 2024) shared task. In this sub-task we had to assign a specific emotion to an utterance which the part of a dialogue. Each episode had multiple speakers speaking in Hinglish. We ranked 11^{th} in this subtask achieving a weighted F1-score of 44%. An end-to-end deep learning pipeline that uses an ensemble of transformer-based Hinglish models was used. We converged on the best models to use in the ensemble by rigorous experimentation using the available models. We also analyse the performance of the classification pipeline and present ablations. We also elaborate on the shortcomings of our systems and some future directions of work.

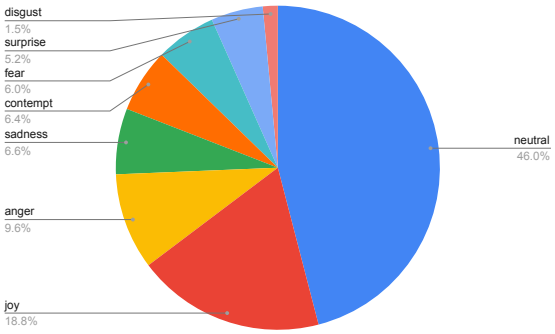


Figure 1: Data Distribution of Training dataset

2 Related Work

Emotion Detection and Sentiment Analysis have been important topics that have been comprehensively studied since the inception of natural language processing. Supervised approaches for Emotion Detection require large datasets which may not be present for low-resource langauges like code-mixed languages.(Orsini, 2015) dates the origin of Hinglish as a language that is widely spoken in India in the post-colonial period. In several works like (Dwivedi and Sukhadeve, 2010), first translation from Hindi-English to English was attempted, however major challenges like non-uniform gram-

* first author, equal contribution
[1] https://github.com/ankit-vaidya19/SemEval24

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 365–369

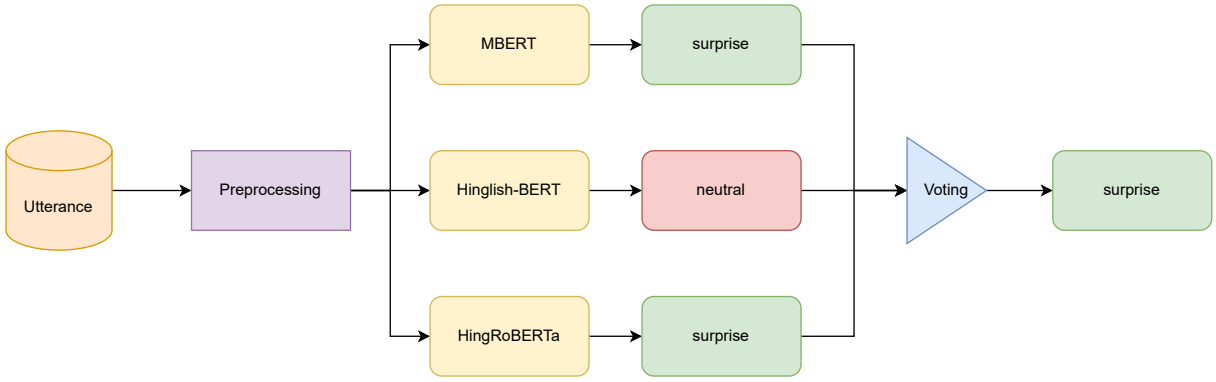


Figure 2: System diagram for Emotion Detection of a sample utterance.

Model	Train F1	Val F1	Test F1
HingBERT	96.72%	45%	43%
HingBERT(LID)	96.67%	44%	42%
HingRoBERTa	96.16%	46%	43%
HingMBERT	96.54%	44%	41%
MBERT	94.76%	41%	40%
Hinglish-BERT	95.76%	42%	41%

Table 1: Comparative results of individual models.

mar and randomised spellings exist could not be overcome.

(Murthy and Kumar, 2021) gives a comprehensive review of modern approaches to detect emotion from text. Extensive work has also been done in the field of Sentiment Analysis of Hinglish text. (Choudhary et al., 2018) made the use of Siamese Networks in order to map the sentences of the code-mixed language and a standard language to a common sentiment space in order to classify the sentences. (Mathur et al., 2018) introduced the Hindi English Offensive Tweets (HEOT) dataset and used a CNN on the embeddings of the data. (Singh and Lefever, 2020) made the use of cross-lingual embeddings obtained using FastText (Bojanowski et al., 2017) and used architectures like CNN, Bi-LSTM and RNN to classify the text. The use of BERT (Devlin et al., 2019) based models was inevitable in this area due to their success in other fields. (Liu et al., 2020) made the use of a pre-trained XLM-RoBERTa (Conneau et al., 2020) and used adversarial examples for the task of sentiment analysis of tweets. However, one thing to note is that most of the prior work has been done on large datasets containing tweets. Due to the large domain shift between analysing tweets and human conversations there was a lack of external training or pre-training data for our task.

For ensemble learning, (Siino et al., 2022) have proposed an ensemble model which generates predictions after the text passes through a vectorisation layer having 2 outputs, one of which is represented as a Bag-of-Words model and provided as input to 3 voters, namely Naive Bayes (NB), Support Vector Machine (SVM) and Decision Tree (DT); and another is a direct input to a CNN. (Kang et al., 2018) proposes a new sentiment analysis method, based on text-based hidden Markov models, that uses word orders without the need of sentiment lexicons. (Miri et al., 2022) proposed use of ensemble feature selection for multi-label text classification which has been used in our approach.

3 Data

The data is in the Hindi-English (Hinglish) code-mixed format which contains words spoken in Hindi but written in the Roman script and English words. The dataset consisted of 343 episodes or dialogues and contained a total of 8,506 utterances which had to classified into eight classes - disgust, contempt, anger, neutral, joy, sadness, fear, surprise. The validation dataset consisted of 46 episodes having 1,354 utterances. The system was then evaluated on a test dataset that contained 57 dialogues consisting of 1,580 utterances. We have illustrated the data distribution in the training dataset in Figure 1. There is acute class imbalance. The class "neutral" contains the most samples (3,123) while the class "disgust" contains the least samples (103). The imbalance ratio was almost 1:30. To mitigate this we tried oversampling to increase size of examples for classes having lower utterances, but they did not improve the performance of the system.

Model 1	Model 2	Model 3	Val F1	Test F1
HingBERT(LID)	HingMBERT	MBERT	45%	42%
HingBERT(LID)	HingBERT	HingMBERT	47%	42%
HingBERT	HingMBERT	MBERT	46%	43%
MBERT	HingMBERT	HingRoBERTa	47%	43%
MBERT	**Hinglish-BERT**	**HingRoBERTa**	**46%**	**44%**
HingMBERT	Hinglish-BERT	HingRoBERTa	46%	44%
HingBERT	MBERT	Hinglish-BERT	44%	44%

Table 2: Results of ensemble pipeline.

4 System Description

The chosen sub-task of emotion detection was a multi-class classification problem which required an utterance to be classified into one of 8 classes. We performed basic pre-processing on the text before passing it to the model. This includes removal of stopwords and punctuation marks from the text, as well as spelling normalisation from the dataset. Due to scarcity of domain specific data related to this task we decided to fine-tune existing transformer-based models to adapt them for our task. Models from (Nayak and Joshi, 2022) like HingBERT, HingRoBERTa, HingMBERT which are based on BERT and RoBERTa (Liu et al., 2019) that were pre-trained on Hinglish data scraped from Twitter were chosen for the task with multilingual models like M-BERT (Devlin et al., 2019). We also chose a variant of BERT (Hinglish-BERT) [2] and a HingBERT variant that was fine-tuned on on the L3Cube-HingLID (Nayak and Joshi, 2022) corpus to include in our system. A linear layer was connected to the pooler output of these models and they were fine-tuned on the dataset. We observed that the performance of the system was enhanced when an ensemble of models was used. We use the method of hard voting to obtain the results from the ensemble. If there was no consensus reached in the ensemble, then the label that the model with the highest F1-score predicted was used as the prediction of the system.

5 Experiments and Results

5.1 Experiments

All the models were used through the HuggingFace (Wolf et al., 2020) library. The data splits that were used during the training and evaluation phase are described in Section 3. The models were fine-tuned for 30 epochs with a learning rate of 1e-5, weight decay of 1e-6 and a batch-size of 32. CrossEntropy loss was used along with the Adam optimizer. We also fixed the seeds to 42. The scoring metric for the task was the weighted F1-score. The scores for the individual models are shown in Table 1. The best performing model checkpoint was chosen according to the epoch-wise validation weighted F1 score. As the individual models had comparable performance on the dataset we decided to create the ensemble by considering all possible combinations of the models. The best performing ensembles and their scores are shown in Table 2.

[2]This model is available here

5.2 Results

The performance of individual models is shown in Table 1 and the performance of the ensemble of models is show in Table 2. The highlighted portion shows our final submission that had a weighted F1-score of 44% consisted of the models MBERT, Hinglish-BERT and HingRoBERTa. We were ranked 11^{th} in the final leaderboard. The difference between our submission and the 5 teams above us was just 1%. We also observed that other combinations also yielded the same result on the dataset as all the models had comparable performance. We also experimented with an ensemble of 5 models (i.e. voting was carried out considering 5 models instead of 3) but the results were similar to our current system and hence, we decided to continue with our current implementation as it is more efficient. The confusion matrix for our submission is illustrated in Figure 3. Note that the confusion matrix has its rows (i.e. true labels axes) normalized according to the number of samples in the class. Here are some observations from our experiments:

1. **The label "anger" has the worst performance:** We observe from Figure 3 that the

label "anger" performs the worst by a significant margin as compared to the rest of the labels despite having relatively more samples compared to some classes. We believe it is due to the fact that the words which characterize anger have a significant overlap with the words that characterize other emotions like "fear" or "contempt".

2. **"joy" vs "surprise" :** We expected the models to confuse these emotions as they are very similar to each other. However, the models rarely confuse these emotions among each other despite the imbalance in the available samples belonging to these two classes. We believe this is due to the fact that these emotions have very distinct appearances in the corpus. We believe that the models captured the subtle difference in the tone that characterize these emotions and thus, could easily differentiate between them.

3. **Failure to capture nuance in negative emotions:** We observe that the overall confusion among negative emotions is higher than the positive emotions. We think that this is due to the fact that many of these emotions have very nuanced differences which the model could not capture due to the scarcity in examples belonging to some of these emotions.

4. **This is a scalable system:** Due to the robust pre-training of the models used, the system could be trained to classify new emotions as well. One could use this system in a continual learning setup in order to increase its capabilities.

6 Conclusion

This paper aims to describe our approach for the ERC sub-task of the 2024 EdiREF Shared Task. We conducted experiments with multiple transformer based models like HingBERT, HingBERT and MBERT. We also show that an ensemble of these models has the best performance on the evaluation dataset with a weighted F1-score of 44%. We foresee several future directions. One direction can be to develop and use more sophisticated methods for ensembling. Another direction is the generation or collection of such data which is more relevant in a real-world scenario in low-resource languages.

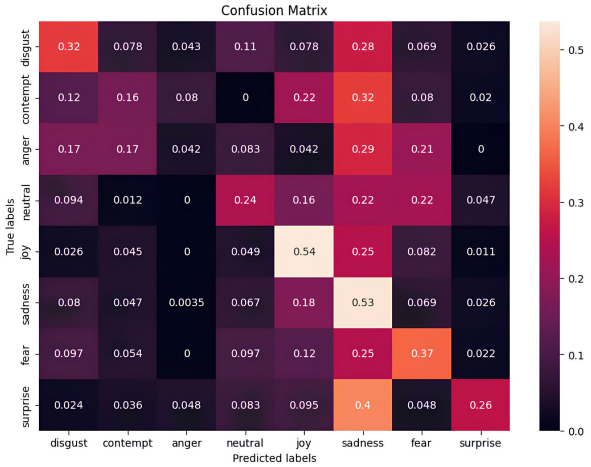


Figure 3: Confusion matrix of system on the Test dataset.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Nurendra Choudhary, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. 2018. Sentiment analysis of code-mixed languages leveraging resource rich languages.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sanjay Dwivedi and Pramod Sukhadeve. 2010. Machine translation system in indian perspectives. *Journal of Computer Science*, 6.

Mangi Kang, Jaelim Ahn, and Kichun Lee. 2018. Opinion mining using ensemble text hidden markov models for text classification. *Expert Systems with Applications*, 94:218–227.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10:

Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Jiaxiang Liu, Xuyi Chen, Shikun Feng, Shuohuan Wang, Xuan Ouyang, Yu Sun, Zhengjie Huang, and Weiyue Su. 2020. Kk2018 at SemEval-2020 task 9: Adversarial training for code-mixing sentiment classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 817–823, Barcelona (online). International Committee for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018. Detecting offensive tweets in Hindi-English code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26, Melbourne, Australia. Association for Computational Linguistics.

Mohsen Miri, Mohammad Bagher Dowlatshahi, Amin Hashemi, Marjan Kuchaki Rafsanjani, Brij B Gupta, and Wadee Alhalabi. 2022. Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*, 37(12):11319–11341.

Ashritha R Murthy and K M Anil Kumar. 2021. A review of different approaches for detecting emotion from text. *IOP Conference Series: Materials Science and Engineering*, 1110(1):012009.

Carol Myers-Scotton. 1993. Duelling languages: Grammatical structure in codeswitching.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Francesca Orsini. 2015. Dil maange more: Cultural contexts of hinglish in contemporary india. *African Studies*, 74(2):199–220.

Marco Siino, Ilenia Tinnirello, Marco La Cascia, et al. 2022. T100: A modern classic ensemble to profile irony and stereotype spreaders. In *CLEF (Working Notes)*, pages 2666–2674.

Pranaydeep Singh and Els Lefever. 2020. LT3 at SemEval-2020 task 9: Cross-lingual embeddings for sentiment analysis of Hinglish social media text. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1288–1293, Barcelona (online). International Committee for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hidetsune at SemEval-2024 Task 4: An Application of Machine Learning to Multilingual Propagandistic Memes Identification Using Machine Translation

Hidetsune Takahashi
Waseda University
takahashi78h@toki.waseda.jp

Abstract

In this system paper for SemEval-2024 Task 4 subtask 2b, I present my approach to identifying propagandistic memes in multiple languages. I firstly establish a baseline for English and then implement the model into other languages (Bulgarian, North Macedonian and Arabic) by using machine translation. Data from other subtasks (subtask 1, subtask 2a) are also used in addition to data for this subtask, and additional data from Kaggle are concatenated to these in order to enhance the model. The results show high reliability of my English baseline and a room for improvement of its implementation.

1 Introduction

SemEval 2024 Task 4 (Dimitrov et al., 2024) calls for classification of memes into different persuasion techniques in textual content only (subtask 1) or in textual and visual content (subtask 2a), and identifying whether or not memes are propagandistic (subtask 2b). I participate in subtask 2b, which is a binary classification problem between "propagandistic" and "non_propagandistic". Various memes are provided in English, Bulgarian, North Macedonian and Arabic to determine whether the memes are propagandistic or not.

My baseline is established so that it achieves fairly high accuracy in English. Although it adopts a classical machine learning method with training, the training data are adjusted by being concatenated with additional dataset. After setting up my baseline, the model is implemented into other languages using machine translation.

Participating in this task allows me to test the ability of my model, achieving a fairly high score for its simplicity. In future studies, the model can be strengthened well enough with appropriately adjusted training data. As for the other languages, on the other hand, the implementation of my English baseline does not necessarily show consistent reliability. Although my baseline works relatively well for Arabic to some extent, the scores go down drastically for Bulgarian and North Macedonian. One of the main reasons for this issue might be that accuracy of machine translation is not high enough, changing the original meanings of the memes and possibly making it more difficult for the English based model to identify propagandistic memes.

My code is available on GitHub [1].

2 Background

The subtask I participate in focuses on classification of memes. Given a json file that has an ID, an image name and text for each meme as input, the subtask requires assigning either "propagandistic" or "non_propagandistic" to the memes. In development phase, data in English were given and participants were allowed to test their solutions for the English data only. Participants were told that they would also have unlabeled test data in three non-English languages as the evaluation phase starts, meaning that no information was released about non-English languages in the development phase.

Propagandistic memes on social media have become a growing issue in the past few years. As more and more people use social media platforms, many types of information including propagandistic one is spread to a number of people (Bradshaw and Howard, 2019). These days, issues caused by propaganda on social media have become worse than most people might think, as a study by O'CONNOR and Weatherall (2019) gives a warning. In fact, these memes did change people's thoughts for elections influencing people's voting behaviors (Aral and Eckles, 2019). Therefore, it might become more and more important in the future for NLP to be able to recognize whether or not the information uses persuasion techniques with high accuracy.

[1] https://github.com/Hidetsune/SemEval2024_Task4.git

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 370–373

There are already many previous studies that adopt modern NLP techniques for propagandistic memes detection. A previous study by Abdullah et al. (2022) uses RoBERTa, which is the state-of-the-art pre-trained language model at the time, resulting in a F1 score of 60.2%. Another previous study by Sprenkamp et al. (2023) tries modern Large Language Models including GPT-3 and GPT-4, also resulting in reliable scores. Al-Omari et al. (2019) utilizes combinations of multiple deep learning models including BERT, BiLSTM and XGBoost with accuracy of around 0.67 in F1 score.

On the other hand of these previous studies, my methodology intends to test classical machine learning approach rather than state-of-the-art LLMs. During my methodology, SpaCy-v3 model (Kömeçoğlu, 2023) is trained with over 20000 memes in English, showing high reliability with accuracy of 0.71353 in F1 macro. Then, the trained model is implemented into non-English languages (Bulgarian, North Macedonian and Arabic) by using Google Translate (Nidhal, 2023).

My participation in this task reveals a high potential of applying classical NLP methods to detection of propagandistic memes with properly adjusted training data. The results reveal that a fairly high score can be achieved without state-of-the-art LLMs and complicated methods. Although the direct implementation of my English baseline into other languages has a room for improvement, the accuracy might go up easily with better machine translation models and careful consideration of differences in topics behind the memes. This paper introduces both strengths and weaknesses of my approach, guiding a direction to future application of classical machine learning to a modern issue of propagandistic memes that requires automatic binary classification.

3 System overview

The main strategy of my system is a classical machine learning method for English baseline and implementation of it into other languages using machine translation. A quick overview of my algorithm is as follows.

1. **Data preparation**: Using official datasets from all the subtasks and additional data on Kaggle[2], training data are prepared to have 11001 memes (including 11000th counting from 0) in English for both propagandistic and non-propagandistic at maximum, making up nearly 22000 memes in total.

2. **Training**: Train a SpaCy model (Kömeçoğlu, 2023) using the prepared training data. Both the training and test data are processed so that they do not have usernames that appear in the additional data and all the memes are lower cased for high efficiency to train the model.

3. **Implementation into non-English languages**: Translate non-English memes into English. This process enables my established baseline to perform in multiple languages, and machine translation is used in this step. After translation of the memes, the model is used in the same way as in English memes.

The system imports a prepared training CSV file as a pandas dataframe, where all the data from Task 4 (train and validation data of subtask 1, subtask 2a and subtask 2b) and additional data from Kaggle[2] are concatenated to compose a large training data with 20774 rows. Only first 20000 rows are used as for the additional data[2] due to its large data size, and all the rows that exceed the limitation of 11001 memes (for both "propagandistic" and "non_propagandistic") are eliminated from the concatenated dataset. Memes in the data are all in English for the purpose of establishing a baseline that guarantees fairly high accuracy in English. Then, SpaCy-v3 model (Kömeçoğlu, 2023) is trained using the resulting data, and test data with unlabeled memes in English is imported as a json file. After that, the trained model is used to assign either "propagandistic" or "non_propagandistic" to each unlabeled meme. As for other languages (Bulgarian, North Macedonian and Arabic), the memes are translated into English so that my English baseline can be implemented into them. Google Translate (Nidhal, 2023) is adopted in this part, and the same trained model is used similarly to the English baseline after translation.

My participation in this task allows for testing the classical NLP approach and simple implementation of it using machine translation. The English baseline, which uses classical machine learning methods, achieves its certain ability to identify propagandistic memes with a reliable score.

On the other hand, the scores go down as for non-English languages. This might be because of

[2] https://www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter

slight changes in meanings in the translation process, which turns out to be the biggest weakness of my approach. Classification of English memes is fairly difficult even for humans given a single utterance. For instance, "VOTE REPUBLICAN. THEY MAY NOT BE PERFECT. BUT THE OTHER SIDE IS INSANE." can be easily classified as propagandistic, but sentences like "CRY ALL YOU WANT..... HE'S DOING EXACTLY WHAT I HIRED HIM FOR....." might not be that easy for the classification because their nuance might depend on situations (on SNS or at work etc.) to some extent. Since propagandistic memes classification is difficult in this way, possible changes of original meanings by machine translation might have resulted in a serious issue for classification of test data, lowering the scores of other languages dramatically.

Another possible reason for the lowered performance in non-English languages is that there are large differences in topics. Words like "Trump" and "Russia" frequently appear in English memes, but "Bulgarian" is one of the most frequently used words in North Macedonian memes. Since Bulgaria and North Macedonia have some diplomatic issues (KAMBERI, 2023), memes that have basis on them ("THE BULGARIANS ENTER THE CONSTITUTION" for example) tend to appear frequently. This kind of differences in topics probably caused the lowered accuracy in non-English languages, revealing a new challenging problem of the approach to implement my English baseline into other languages.

4 Experimental setup

Before moving on to the actual training of the model, data preparation was an essential part of my solution. First of all, all the data of this task (including other subtasks) are imported as json files and converted into pandas dataframes. The dataframes have "text" and "propagandistic/non_propagandistic" as columns. Training data and validation data from both subtask 1 and subtask 2 are composed of "propagandistic" only, so they are concatenated as "large_data1", which is a dataframe with 8307 rows and all "propagandistic" memes. Also, data from Kaggle[2] are imported as all "non_propagandistic" dataframe and processed so that it has no usernames in "text" column. Validation data of subtask 2b are previously imported, but I decided to increase weight of training data of task 2b by twice and not to use validation data after trial submissions in development phase. Therefore, the resulting training data are composed of training data from subtask 2b (weight increased by twice), "large_data1" and additional data from Kaggle[2]. The training dataframe is adjusted so that it has 11001 memes for both propagandistic and non-propagandistic at maximum, being shuffled to be ready for training.

The reason why the additional data from Kaggle[2] are chosen is that they are less likely to be propagandistic compared to many other existing datasets. The data are from customer support on Twitter including AppleSupport and AmazonHelp, so the topic there should be something related to their products or services. There are many other existing datasets extracted from social media platforms, but it takes too much time end effort to manually assign propagandistic and non-propagandistic to each utterances. For the purpose of getting non-propagandistic utterances only, it might be one of the easiest and most realistic approach to find an existing dataset whose topic is clearly unrelated to politics and diplomacy as included in my solution.

After these data preparation steps, the model (Kömeçoğlu, 2023) is trained with the training dataset, and test data with unlabeled memes are imported as a json file. The memes in non-English languages are translated into English as stated in the previous section. The trained model assigns either "propagandistic" or "non_propagandistic" to each meme. The training data and test data are cleaned with new line removal and lower casing prior to use of them.

LANGUAGE	F1 macro	F1 micro	Ranking
English	0.71353	0.79000	13/20
Bulgarian	0.32670	0.33000	14/14
North Macedonian	0.38942	0.46000	13/14
Arabic	0.52825	0.54375	9/14

Table 1: Task scores for multiple languages

5 Results

Table 1 shows official results of my solutions. They show fairly high accuracy of my English baseline with nearly 0.8 in F1 micro. It can be said that my methodology for English baseline using classical machine learning works fairly well with a thoughtful training data adjustment.

As for the non-English languages, the scores go

down due to the potential reasons as stated in prior sections. Even so, Arabic has relatively high accuracy, which is lower than English by around 0.18 in F1 macro but higher than Bulgarian by around 0.20. This difference might have been caused by accuracy of machine translation mainly. Arabic is a widely used language with a total of about 372.7 million native speakers in the world.[3] There is a possibility that Google Translate (Nidhal, 2023), which I use for machine translation, has higher accuracy for widely used languages including Arabic, maintaining the original meanings and nuances of the memes fairly correctly.

6 Conclusions

To summarise, my methodology firstly focuses on establishing a baseline that guarantee fairly high accuracy for English memes. After that, the baseline is implemented into non-English languages by translating the memes into English using machine translation.

The results show high reliability of my English baseline. The methodology for the baseline has its basis on classical machine learning, but my participation in this task reveals its fundamental abilities to deal with complicated classification task with properly adjusted training data.

On the other hand, the results also show that the simple application of my English baseline has a room for improvement. The scores of non-English languages dramatically dropped although Arabic has relatively reasonable accuracy compared to Bulgarian and North Macedonian. There can be many possible reasons for this including the accuracy of machine translation and changes in topics between memes in different languages.

In future studies, it might be worthwhile to enhance the model with many more memes for my English baseline. Collecting propagandistic memes might be a time consuming task, but non-propagandistic memes, on the other hand, can be easily found and used by utilizing existing datasets whose topics clearly have nothing to do with politics and diplomacy. As for non-English languages, higher accuracy might be achieved by using better machine translation models and enhancing the baseline model with specific political or diplomatic topics in the countries.

[3] https://www.worlddata.info/languages/arabic.php

References

Malak Abdullah, Ola Altiti, and Rasha Obiedat. 2022. Detecting propaganda techniques in english news articles using pre-trained transformers. In *2022 13th International Conference on Information and Communication Systems (ICICS)*, pages 301–308.

Hani Al-Omari, Malak Abdullah, Ola AlTiti, and Samira Shaikh. 2019. JUSTDeep at NLP4IF 2019 task 1: Propaganda detection using ensemble deep learning models. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 113–118, Hong Kong, China. Association for Computational Linguistics.

Sinan Aral and Dean Eckles. 2019. Protecting elections from social media manipulation. *Science*, 365(6456):858–861.

Samantha Bradshaw and Philip N Howard. 2019. The global disinformation order: 2019 global inventory of organised social media manipulation.

Dimitar Dimitrov, Giovanni Da San Martino, Preslav Nakov, Firoj Alam, Maram Hasanain, Abul Hasnat, and Fabrizio Silvestri. 2024. SemEval-2024 Task 4: MULTILINGUAL DETECTION OF PERSUASION TECHNIQUES IN MEMES. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.

Donika KAMBERI. 2023. An overview of the dispute between north macedonia and bulgaria through the optic of international law. *JUSTICIA–International Journal of Legal Sciences*, 11(19-20):69–75.

Başak Kömeçoğlu, Buluz. 2023. Emotion Classification with SpaCy v3 Comet.

Baccouri Nidhal. 2023. [The article referred to for machine translation]. *pypi*.

CAILIN O'CONNOR and James Owen Weatherall. 2019. The social media propaganda problem is worse than you think. *Issues in Science and Technology*, 36(1):30–32.

Kilian Sprenkamp, Daniel Gordon Jones, and Liudmila Zavolokina. 2023. Large language models for propaganda detection. *arXiv preprint arXiv:2310.06422*.

Hidetsune at SemEval-2024 Task 10: An English Based Approach to Emotion Recognition in Hindi-English code-mixed Conversations Using Machine Learning and Machine Translation

Hidetsune Takahashi
Waseda University
takahashi78h@toki.waseda.jp

Abstract

In this system paper for SemEval-2024 Task 10 subtask 1 (ERC), I present my approach to recognizing emotions in Hindi-English code-mixed conversations. I train a SpaCy model with English translated data and classify emotions behind Hindi-English code-mixed utterances by using the model and translating them into English. I use machine translation to translate all the data in Hindi-English mixed language into English due to an easy access to existing data for emotion recognition in English. Some additional data in English are used to enhance my model. This English based approach demonstrates a fundamental possibility and potential of simplifying code-mixed language into one major language for emotion recognition.

1 Introduction

SemEval 2024 Task 10 (Kumar et al., 2024) calls for assigning emotions to Hindi-English code-mixed conversations (ERC) and reasoning emotion flips (EFR) in Hindi-English code-mixed conversations and in English conversations. I participate in subtask 1 (ERC), for which Hindi-English code-mixed utterances are given in text for participants to recognize emotions behind them. An emotion from disgust, contempt, anger, neutral, joy, sadness, fear and surprise is assigned to each utterance as the correct emotion associated with it.

My methodology has its basis on emotion detection in English. I translate all the development data into English using machine translation (Adep, n.d.), and use the data to train a SpaCy model (Kömeçoğlu, 2023). I also use data by Nikam (Nikam, n.d.) to enhance my model, where utterances in English and their previously assigned emotions are given as a CSV file. He mainly introduces his own model, but I made use of the data only.

I leverage an easy access to emotion-assigned data in English and its linguistic simplicity. Therefore, the point of my approach is to test whether or not machine translation of Hindi-English code-mixed conversations into English can contribute to fundamental accuracy for emotion recognition with lower complexity and an easier access to additional data.

The result shows that combination of machine translation and machine learning works with reasonable accuracy for emotion detection considering its simplicity and numerous data in English that can be implemented in future studies.

My code is available on GitHub [1].

2 Background

The subtask I participate in focuses on emotion recognition in Hindi-English code-mixed conversations. Given Hindi-English code-mixed utterances ("kuchh karo sahil please kuchh karo. mera rosesh adopt ho karke chala gaya na to me, i know this sounds horribly melodramatic, monishaish, par me mar jaaungi. i swear mein mar jaaungi" for example) as input, the subtask requires assigning an emotion to each of them as output. Episode name and speakers' names are given as input as well, but they are not used for my solutions.

Before proceeding with this task, Hindi-English code-mixed language needs to be explained in detail. Hindi-English code-mixed language, which is often referred to as *Hinglish*, is a language in which speakers mix Hindi and English in conversations. According to a study by Chand (2016), there are some Hinglish speakers who cannot speak pure Hindi, and even those who speak both Hindi and Hinglish tend to speak Hinglish in conversations with monolingual speakers of Hinglish, causing the number of monolingual speakers of Hinglish to grow as a result. Therefore, the situation in this task is not a small topic but rather an essential one for the future Indian communities.

[1] https://github.com/Hidetsune/SemEval2024_Task10.git

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 374–378

Most previous studies aim to recognize emotions in Hinglish by collecting Hinglish sentences and using them directly for their approaches. A study by Vijay et al. (2018) uses n-grams for their solution with Hinglish on Twitter. The work detects emotions with high accuracy, but they state that the accuracy drops by nearly 16% without char n-grams. Another previous work by Sasidhar et al. (2020), which uses the solution by Vijay et al. (2018) as their baseline, collects more data with 12000 Hindi-English code-mixed sentences for training. In addition to these, a study by Wadhawan and Aggarwal (2021) achieves high accuracy with a transformer based BERT model, and another study by Kaur et al. (2019) deals with Hinglish on YouTube comment sections. In all of these works, they process Hindi-English code-mixed sentences without translating them into other languages.

On the other hand, one of the biggest issues with emotion recognition in Hinglish conversations might be shortage of datasets. Although there are many Hinglish sentences expressing emotions on social media platforms, the language they use on them might differ from in real conversations to some extent. Taking the availability of numerous existing datasets in English into account as well, I decided to explore a solution to use English translated data rather than the original Hinglish data. This trial requires machine translation, and this step is combined with classical machine learning method, achieving reasonable accuracy for its simplicity with an additional English dataset concatenated to the translated dataset.

Emotion recognition in Hinglish can be complicated because the language has technically two languages (Hindi+English). However, when simplified into all in English properly, there is much more potential to deal with Hinglish emotion recognition with an easy access to enormous English datasets and established NLP methods that have been mainly used for English. This paper aims to guide a direction to future application of this approach, establishing the basis with easily used datasets and model.

3 System overview

The main strategy of my system is a combination of classical machine learning method with simplification of Hindi-English code-mixed sentences into English sentences. Therefore, my methodology is composed of data preparation using machine translation and the classic machine learning process with training. A quick overview of my algorithm is as follows.

1. **Official training data and development data translation**: The official data in json file format are imported and converted into pandas dataframes. The dataframes have episode names, utterances, speaker names and emotions as columns. The utterances are translated into English and saved in a new column.

2. **Additional data concatenation**: Concatenate additional data, which have utterances in English and the emotions, with the translated official data.

3. **Addressing data imbalance**: Separate the concatenated data into each emotion and set a limitation of 3001 utterances (including 3000th counting from 0) per one emotion type.

4. **Model training and prediction**: Train a model with re-concatenated data to predict emotions for unlabeled evaluation data.

For data preparation, I use the official training and development dataset (Kumar et al., 2023), and additional data by Nikam (n.d.) to enhance my model and mitigate data imbalance. Firstly, I import the official datasets in json file format (Kumar et al., 2023), and convert them into pandas dataframes which have episode names, utterances in Hindi-English mixed language (pronunciation forms of Hindi + English), speaker names and emotions as columns. Then, I use Google Translate (Adep, n.d.) to translate all the utterances in Hindi-English mixed language into English, and I add the translated sentences to the dataframe as a new column named "utterances_English". Since episode name, utterances in mixed language and speaker names are of no use at this time, they are dropped from the dataframe to have only "emotion" and "utterances_English" as columns on the dataframes. After that, they are concatenated into one dataframe. There is a huge data imbalance and shortage of training data at this point, so I use additional data by Nikam (n.d.) to mitigate these problems.

In the next step, I train the SpaCy-v3 model (Kömeçoğlu, 2023) with the prepared data and use it to assign emotions to unlabeled evaluation data. The evaluation data are composed of episode

name, utterances in Hindi-English mixed language, speaker names as columns. "utterances_English" column, in which machine translated sentence is assigned to each given utterance, is added to have the model predict emotions behind the utterances.

Participating in this emotion recognition task using the combined strategy enables me to explore the potential of implementing machine translation into a specific situation as this task. The application of machine translation to machine learning makes it possible for a multi-label text classifier to predict emotions behind Hinglish sentences with reasonable accuracy for the entry of this trial. Considering the saved complicated steps needed to handle the data as text in two separate languages, you can see the potential of simplification that my methodology aims at by combing machine translation with machine learning. Future applications for higher accuracy might include more training data in Hindi-English mixed sentences and enhancement of the model with the implementation of machine translation into Hindi written sentences. Participating in this task reveals both strengths and weaknesses of my strategy, guiding directions of future studies to apply machine translation and classic NLP methods to emotion recognition in Hinglish conversations.

4 Experimental setup

Before moving on to the actual training of the model, some setups were required to prepare training data. As stated before, the utterances in the provided development data (Kumar et al., 2023) are all in Hindi-English mixed language. Since my participation in this task intends to combine machine translation with machine learning for emotion recognition, I decided to simplify the data by translating them into one language rather than taking the complexity of separating them into the two languages. This simplification by machine translation might have changed the original meanings of the utterances, but the difference might not be significant for recognizing the emotions only. Looking through the Hindi-English code-mixed sentences, I noticed that fairly large parts of the utterances are in pronunciation forms in Hindi and that English is used only partly. For instance, some short words or phrases like "goodbye!" are utterances where English appears only, but sentences like "lekin what about my ghadi? 17000 ki ghadi hai..." are mostly composed of pronunciation forms of Hindi (except "what about my" in this case). Pronunciation forms of Hindi are much more prevalent in many other utterances.

There could be two possible choices in my approach; translate all the mixed language sentences (Hindi+English) into Hindi or into English. It might be a good idea as well to choose the former considering the high prevalence of Hindi, but I chose the latter due to the higher availability of reliable additional data on emotion recognition in English.

As for machine translation, Google Translate (googletrans 3.1.0a-0) was used. The reference (Adep, n.d.) uses it to translate sentences in all actual Hindi characters (no English and in written form of Hindi; not in pronunciation form). On the other had, I undertook an experiment to try it for the official development dataset (Hindi-English code-mixed and pronunciation form for Hindi; not in written form) and founded that it works. Therefore, all the given utterances are translated into English and added to the dataset as a new column.

At the next step, additional data are collected and processed to be concatenated with the translated version of official data. Since there is a huge data imbalance and lack of training data as stated in the previous section, it is obvious that additional data is needed where most of the 8 emotions used in this task (disgust, contempt, anger, neutral, joy, sadness, fear and surprise) are labeled for utterances or sentences. For consistency of additional data, one source was used rather than concatenating multiple sources with different categories of emotions.

Data by Nikam (Nikam, n.d.) were chosen for additional data. The original concatenated data (the official training and validation data) are composed of text in English and 8 emotions (disgust, anger, neutral, joy, sadness, fear, contempt and surprise) for each utterance. The only two differences in categories of emotion are that the additional data does not have "contempt" while the official one have, and that the additional data have "shame" while the official one does not. Details are as shown on Table1. The concatenated version of official data and the additional data are concatenated separately for each emotion, and the number of utterances is adjusted after that. Conducting some experimental trial on development phase, I decided to limit the data so that it has 3001 utterances (3000th rows counting from 0) at maximum for each emotion type, and all the utterances that exceed the number (from 3002th rows) are dropped from the training

dataset.

5 Results

In the evaluation phase, my trained model worked with accuracy of 0.39 (weighted F1 score calculated by the organizers' system), and the ranking was 17th out of 39 participants. The result is not as good as to use for a practical use as of now, but it definitely shows a basic ability of my approach to apply machine translation to Hindi-English code-mixed language.

There are certain weaknesses in my algorithm as the result shows. One of the biggest weaknesses is the lack of dataset due to inevitable data imbalance. Concatenating the official data (Kumar et al., 2023) with additional data (Nikam, n.d.), there are only 1004 utterances associated with disgust for example while there are many more utterances than 3001 for anger. Limiting the number of utterances up to 1001 for each emotion yielded low accuracy of 0.27 in the development phase, so I chose to accept data imbalance to some extent with the limitation of 3001 utterances for each emotion so that the model is trained better. Larger data imbalance was inevitable to make use of more data, yielding lower scores in development phase, so I had no choice but to decide on around that limitation. However, it cannot be said that less than around 3000 training utterances per emotion are enough for accurate emotion recognition, which is one of the main weaknesses of my solution in this task.

In addition to that, the process of machine translation might have changed the original meaning of the utterances, which might have lowered the quality of training. I cannot look deeper into this possible issue because I am not a Hindi speaker, but the accuracy might go up with a better machine translator.

Another weakness of my solution is that the additional data I used (Nikam, n.d.) are provably not from actual conversations. There are many "@" followed by what I suppose are usernames, so the entire additional data (Nikam, n.d.) are probably from social media platforms. Since this task deals with text version of conversations, situations of the datasets are probably unmatched with each other.

On the other hand, the strength of my approach is the linguistic simplicity that enables the model to have its potential to utilize established NLP techniques and numerous data that have been developed for English. Since English is used widely around the world, there are tons of other data sources that can be implemented into my model. It goes without saying that there are already many existing data in which emotions are labeled with sentences, I can also write down daily conversations in English into text and label each utterance with an emotion for higher accuracy since this task deals with data from conversations.

6 Conclusions

To summarize, I firstly simplified utterances in Hindi-English code-mixed language by translating them into English. Machine translation (Adep, n.d.) is used in this step, and additional data (Nikam, n.d.) are concatenated to mitigate data imbalance and enhance the model. Utterances of evaluation data (Kumar et al., 2023) are also translated into English by machine translation in the same way, and the trained model predicted an emotion for each utterance.

Although my solution has a room for improvement, the result shows a basic ability of the simplification by machine translation to recognize emotions behind Hindi-English code-mixed utterances. Given the abundance of data and established NLP techniques in English, my approach, combining machine translation with classical NLP methods, might open the door for addressing the challenges in emotion recognition in Hinglish caused by its linguistic complexity.

References

Venugopa Adep. n.d. Hindi to English translation using Python. *kaggle*.

Vineeta Chand. 2016. The rise and rise of Hinglish in India.

Gagandeep Kaur, Abhishek Kaushik, and Shubham Sharma. 2019. Cooking is creating emotion: A study on hinglish sentiments of youtube cookery channels using semi-supervised approach. *Big Data and Cognitive Computing*, 3(3).

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense

Dataset	Anger	Contempt	Disgust	Fear	Joy	Neutral	Sadness	Surprise	Shame
Concatenated official data	937	616	148	602	1824	4542	684	507	0
Additional data	4286	0	856	5409	11037	1811	6719	4062	146

Table 1: Datasets and emotion categories

to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Başak Kömeçoğlu, Buluz. 2023. Emotion classification with SpaCy v3 and comet.

Sanket Nikam. n.d. Emotion detection in text using natural language processing.

T Tulasi Sasidhar, Premjith B, and Soman K P. 2020. Emotion detection in Hinglish(Hindi+English) code-mixed social media text. *Procedia Computer Science*, 171:1346–1352. Third International Conference on Computing and Network Communications (CoCoNet'19).

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Corpus creation and emotion prediction for Hindi-English code-mixed social media text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 128–135, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in hindi-english code-mixed data: A transformer based approach. *arXiv preprint arXiv:2102.09943*.

All-MPNet at SemEval-2024 Task 1: Application of MPNet for Evaluating Semantic Textual Relatedness

Marco Siino
Department of Electrical, Electronic
and Computer Engineering
University of Catania
Italy
marco.siino@unipa.it

Abstract

In this study, we tackle the task of automatically discerning the level of semantic relatedness between pairs of sentences. Specifically, Task 1 at SemEval-2024 involves predicting the Semantic Textual Relatedness (STR) of sentence pairs. Participants are tasked with ranking sentence pairs based on their proximity in meaning, quantified by their degree of semantic relatedness, across 14 different languages. To each sentence pair is assigned a manually determined relatedness score ranging from 0 (indicating complete lack of relation) to 1 (denoting maximum relatedness). In our submitted approach on the official test set, focusing on Task 1 (a supervised task in English and Spanish), we achieve a Spearman rank correlation coefficient of 0.808 for the English language and of 0.611 for the Spanish language.

1 Introduction

The notion of semantic relatedness between language units has been foundational in understanding meaning. Automatic determination of relatedness has wide-ranging applications, including assessing sentence representation methods, question answering, and summarization (Guarino, 1997).

Two sentences are deemed semantically similar when they exhibit a relationship of entailment or paraphrases. In contrast, relatedness encompasses a broader concept, encompassing all commonalities between two sentences: whether they pertain to the same topic, convey the same viewpoint, stem from the same temporal context, one elaborates on the other, and so forth (Hadj Taieb et al., 2020).

Historically, much of NLP research has focused on semantic similarity, predominantly in English. However, in the shared Task 1 hosted at SemEval 2024 (Ousidhoum et al., 2024b,a), the organizers extend their coverage to include the following languages: Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu.

With the advancement of machine and deep learning architectures in recent years, there has been a surge of interest in NLP. Numerous efforts have been dedicated to creating algorithms capable of automatically identifying and categorizing text information available on the internet. In the literature, several strategies have already been proposed. In the last fifteen years, some of the most successful strategies have been based on SVM (Colas and Brazdil, 2006; Croce et al., 2022), on Convolutional Neural Network (CNN) (Kim, 2014; Siino et al., 2021), on Graph Neural Network (GNN) (Lomonaco et al., 2022), on ensemble models (Miri et al., 2022; Siino et al., 2022) and, recently, on Transformers (Vaswani et al., 2017; Siino et al., 2022).

The increasing adoption of Transformer-based architectures in academic research has also been bolstered by various methodologies showcased at SemEval 2024. These methodologies tackle diverse tasks and yield noteworthy findings. For instance, at the Task 2 (Jullien et al., 2024), where to address the challenge of identifying the inference relation between a plain language statement and Clinical Trial Reports is used T5 (Siino, 2024c); Task 4 (Dimitrov et al., 2024) where is employed a Mistral 7B model to detect persuasion techniques in memes (Siino, 2024b); and Task 8 (Wang et al., 2024), that utilizes a DistilBERT model to identify machine-generated text (Siino, 2024a).

For our model development, we devised a two-stage architecture. In the first stage, we utilized a Sentence Transformer specifically trained in a multilingual domain. Subsequently, we computed the cosine similarity of the generated embeddings to predict the relatedness between the analyzed sentences.

The remainder of this paper is structured as follows: Section 2 provides background information

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 379–384

on Task 1 hosted at SemEval-2024. Section 3 presents an explanation of the submitted approach. We detail the experimental setup required to reproduce our work in Section 4. The results of the formal assignment and pertinent discussions are presented in Section 5. Finally, we conclude with our findings and suggestions for future research in Section 6.

We make all the code publicly available and reusable on GitHub[1].

2 Background

Data for Semantic Textual Relatedness (STR) Shared Task 1[2] includes sentence pairs labeled with scores representing the degree of semantic textual relatedness between them, ranging from 0 (completely unrelated) to 1 (maximally related). These scores have been determined through manual annotation using a comparative annotation approach to mitigate biases commonly associated with traditional rating scale methods. This annotation process ensures a high reliability of the relatedness rankings.

The task involves predicting the STR of sentence pairs in 14 different languages. Participating teams were asked to submit systems for one, two, or all of the following tracks:

- Track A: Supervised — Systems trained using labeled training datasets provided. Additional publicly available datasets can be used, but teams must report the additional data and its impact on results.

- Track B: Unsupervised — Systems developed without using labeled datasets related to semantic relatedness or similarity between text units longer than two words in any language. Use of unigram or bigram relatedness datasets is permitted.

- Track C: Cross-lingual — Systems developed without using labeled semantic similarity or relatedness datasets in the target language, but with the use of labeled dataset(s) from at least one other language. Using labeled data from another track is mandatory for submissions to this track.

[1] https://github.com/marco-siino/SemEval2024/tree/main/Task%201

[2] https://semantic-textual-relatedness.github.io/

3 System Overview

The illustration of the proposed approach is provided in the Figure 1. Upon selecting a sample (i.e., a pair of sentences) from the dataset, the initial step involves encoding the first sentence using the All-MPNet embedding, thereby generating an embedding vector. Subsequently, employing an identical procedure, the second sentence from the sample is also encoded. The resulting embedding vectors are then subjected to a cosine similarity computation, facilitating the derivation of the semantic similarity prediction between two sentences.

To develop our model, we thought of a two-stage architecture. In the first stage, we used a *Sentence Transformer*. This is a Python framework for cutting-edge sentence, text, and image embeddings. The initial work is described in (Reimers and Gurevych, 2019). More than 100 languages have sentences and text embeddings that can be computed using this method. Sentences with a similar meaning can subsequently be found by comparing these embeddings, for example, using cosine-similarity. Semantic search, paraphrase mining, and semantic textual similarity can all benefit from these embeddings. The framework offers a huge selection of pre-trained models suited for different tasks and is built on PyTorch and Transformers. Moreover, fine-tuning models is also feasible.

The model used as Sentence Transformer is *all-mpnet-base-v2*, and it is available on HuggingFace[3]. The model is based on MPNet (Song et al., 2020). MPNet introduces a novel pre-training approach that combines the strengths of BERT and XLNet while addressing their respective limitations. Unlike BERT's masked language modeling (MLM), MPNet utilizes permuted language modeling (PLM) to capture dependencies among predicted tokens more effectively. Additionally, MPNet incorporates auxiliary position information as input, allowing the model to process full sentences and mitigate position discrepancy issues present in XLNet. Pre-training of MPNet is conducted on a large-scale dataset exceeding 160 GB of text corpora, followed by fine-tuning on various downstream tasks such as GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016). Experimental findings demonstrate that MPNet significantly outperforms both MLM and PLM, achieving superior results across these tasks compared to previ-

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Figure 1: Block diagram of our proposed approach. Given a sample from the dataset, the first sentence is encoded with an All-MPNet embedding for generating an embedding vector as output. Also, the second sentence in the sample is encoded in the same way. Then the two embedding vectors are compared using cosine similarity to produce the final prediction on the semantic similarity of the two sentences.

ous state-of-the-art pre-trained models like BERT, XLNet, and RoBERTa, all under the same model configuration.

The model was used to map all the words present in the text to a word embedding space. Following the embeddings, the cosine distance between two sentences was calculated. The cosine similarity between the two embedding vectors is calculated as shown in the Equation 1.

$$\cos(\theta) = \frac{A \cdot B}{\|A\|_2 \|B\|_2} \quad (1)$$

The value provided as cosine similarity was then provided as the requested prediction for the two sentences considered. Our code is available online together with the predictions generated and sent in relation to the test set.

The recent study by (Siino et al., 2024b) highlights that preprocessing for text classification tasks lacks significant impact when employing Transformers. Specifically, the study finds that the optimal preprocessing strategies do not substantially differ from performing no preprocessing at all, particularly in the case of Transformers. Consequently, in order to maintain our system's efficiency, speed, and computational lightness, we have opted to not conduct any preprocessing on the text. This decision aligns with the findings of the study and underscores the effectiveness of Transformers in handling raw text data without the need for extensive preprocessing steps.

4 Experimental Setup

We implemented our model on Google Colab[4]. The library we used was Sentence Transformer. The library requires Python[5] (>= 3.8) and PyTorch[6] (>=1.11.0). The dataset provided for all the phases are available on the official competition page. On the basis of our preliminary experiments, we found beneficial to set the threshold value upon the cosine similarity equal to 0.5. We did not perform any additional fine-tuning on the MPNet embeddings. To run the experiment, a T4 GPU from Google has been used. After the generation of the predictions, we exported the results on the JSON format required by the organizers. As already mentioned, all of our code is available on GitHub.

5 Results

The official evaluation metric for this task is the Spearman rank correlation coefficient, which evaluates how closely the rankings predicted by the system align with human judgments. The evaluation script for this shared task is available on the GitHub page, providing a standardized method for assessing the performance of participating systems. The formula to compute the Spearman correlation coefficient is provided in the Equation 2.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (2)$$

Where d represents the pairwise distances of the ranks of the variables x_i and y_i, while n is the number of samples.

In Table 1, are reported the results obtained on the two languages we considered for our participation at the task. Considered the very low effort required to run the proposed approach and to generate the predictions, the score of 0.611 and 0.808 appears to be an interesting baseline, while still exhibiting room for improvements. It is worth noticing that the approach is a Zero-Shot one with no prior knowledge on the specific task.

In the Table 2 and 3, the results obtained by the first three teams and by the last one, as showed on the official CodaLab page, are reported. Furthermore, we reported the baselines for the two languages. Compared to the best performing models, our simple approach exhibits some room for improvements. However, it is worth notice that it

[4] https://colab.research.google.com/
[5] https://www.python.org/
[6] https://pytorch.org/

LANGUAGE	Score
English	0.808
Spanish	0.611

Table 1: The suggested method's performance on the test set. Our results are related to our participation in the Track A, for the English and for the Spanish languages only.

TEAM NAME	Score
PALI (1)	0.859
UAlberta (2)	0.853
Tübingen-CL (3)	0.850
SemRel-SemEval Baseline (*)	0.830
YNUNLP2023 (36)	0.557

Table 2: Comparing performance on the test set for the English language. In the table are shown the results obtained by the first three users and by the last one. In parentheses is reported the position in the official ranking.

required no further pre-training and the computational cost to address the task is manageable with the free online resources offered by Google Colab.

Even if our approach is simple and straightforward, here we want to qualitative analyze some results obtained with our approach to better motivate some classification mistakes. With regard to the English language, the first relevant misclassification sample is related to the sample pair: "This book is very compelling. – best book so far in the series!". Our system predicts a cosine similarity equal to 0.47 while the actual similarity is 0.73. In this case, in fact, it is very hard to assess if "being very compelling" is so semantically similar to the concept of "being the best so far in a series". Furthermore, looking at the sample: "A woman in a black coat eats dinner while her dog looks on. – A little boy is standing on the street while a man in overalls is working on a stone wall.", the prediction using our approach is equal to 0.0 (no semantic similarity) while the actual target for the provided test set is 0.29. Another interesting case — i.e., our approach predicts a high similarity of 0.92 while the actual target is 0.74 — is given by the sample: "My favorite by far is definitely Chris and I think he will win!! – My favorite and the selection for winner is Chris.". From a semantic perspective, however, both the sentences provide the same two concepts (i.e., Chris is my favorite, I think he will win). Given these and several others differences between our predictions and the actual target similarity in the provided test set, some concerns on the labelling process and on the correctness of the provided target similarity values can be raised.

TEAM NAME	Score
AAdaM (1)	0.740
GIL-IIMAS UNAM (2)	0.731
PALI (3)	0.724
SemRel-SemEval Baseline (*)	0.7
YNUNLP2023 (25)	0.404

Table 3: Comparing performance on the test set for the Spanish language. In the table are shown the results obtained by the first three users and by the last one. Furthermore, the baseline is also provided. In parentheses is reported the position in the official ranking.

6 Conclusion

This paper introduces the utilization of an All-MPNet model embedding to tackle Task 1 at SemEval-2024. In our submission, we opted for a straightforward Zero-Shot learning approach, leveraging pre-trained Transformers that are already tailored to a multilingual-domain. Following this approach, we utilized the contextual embeddings generated by the Sentence Transformer, and we employed cosine distance to measure the similarity between pairs of sentences, thus quantifying the STR between them. Despite the effectiveness of our method, there remains room for improvement, as indicated by the final ranking. Potential alternative approaches could involve leveraging the zero-shot capabilities of models such as GPT and T5, expanding the training data size by incorporating additional datasets, or exploring different methods of integrating ontology-based domain knowledge into our approach. Furthermore, given the interesting results recently provided on a plethora of tasks, also few-shot learning (Wang et al., 2023; Maia et al., 2024; Siino et al., 2023; Meng et al., 2024) or data augmentation strategies (Muftie and Haris, 2023; Siino et al., 2024a; Tapia-Téllez and Escalante, 2020; Siino and Tinnirello, 2023) could be employed to improve the performance. Compared to the best performing models, our simple approach exhibits some room for improvements. However, our qualitative analysis raised some concerns on the labels provided for the test set. Then, we are not able to correctly assess the actual performance of our proposed approach. Eventually, it

is worth notice that thanks to our approach no further pre-training is required and the computational cost to address the task is manageable with the free online resources offered by Google Colab.

Acknowledgments

We would like to thank anonymous reviewers for their comments and suggestions that have helped to improve the presentation of the paper.

References

Fabrice Colas and Pavel Brazdil. 2006. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

Daniele Croce, Domenico Garlisi, and Marco Siino. 2022. An SVM ensemble approach to detect irony and stereotype spreaders on twitter. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2426–2432. CEUR-WS.org.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Nicola Guarino. 1997. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology: International Summer School, SCIE-97 Frascati, Italy, July 14–18, 1997*, pages 139–170. Springer.

Mohamed Ali Hadj Taieb, Torsten Zesch, and Mohamed Ben Aouicha. 2020. A survey of semantic relatedness evaluation datasets and procedures. *Artificial Intelligence Review*, 53(6):4407–4448.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Francesco Lomonaco, Gregor Donabauer, and Marco Siino. 2022. COURAGE at checkthat!-2022: Harmful tweet detection using graph neural networks and ELECTRA. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 573–583. CEUR-WS.org.

Beatriz Matias Santana Maia, Maria Clara Falcão Ribeiro de Assis, Leandro Muniz de Lima, Matheus Becali Rocha, Humberto Giuri Calente, Maria Luiza Armini Correa, Danielle Resende Camisasca, and Renato Antonio Krohling. 2024. Transformers, convolutional neural networks, and few-shot learning for classification of histopathological images of oral cancer. *Expert Systems with Applications*, 241:122418.

Zong Meng, Zhaohui Zhang, Yang Guan, Jimeng Li, Lixiao Cao, Meng Zhu, Jingjing Fan, and Fengjie Fan. 2024. A hierarchical transformer-based adaptive metric and joint-learning network for few-shot rolling bearing fault diagnosis. *Measurement Science and Technology*, 35(3).

Mohsen Miri, Mohammad Bagher Dowlatshahi, Amin Hashemi, Marjan Kuchaki Rafsanjani, Brij B Gupta, and W Alhalabi. 2022. Ensemble feature selection for multi-label text classification: An intelligent order statistics approach. *International Journal of Intelligent Systems*, 37(12):11319–11341.

Fuad Muftie and Muhammad Haris. 2023. Indobert based data augmentation for indonesian text classification. In *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, page 128 – 132.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marco Siino. 2024a. Badrock at semeval-2024 task 8: Distilbert to detect multigenerator, multidomain and multilingual black-box machine-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024b. Mcrock at semeval-2024 task 4: Mistral 7b for multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino. 2024c. T5-medical at semeval-2024 task 2: Using t5 medical embeddings for natural language inference on clinical trial data. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2021. Detection of hate speech spreaders using convolutional neural networks. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 2126–2136. CEUR-WS.org.

Marco Siino, Marco La Cascia, and Ilenia Tinnirello. 2022. Mcrock at semeval-2022 task 4: Patronizing and condescending language detection using multi-channel cnn, hybrid lstm, distilbert and xlnet. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 409–417. Association for Computational Linguistics.

Marco Siino, Francesco Lomonaco, and Paolo Rosso. 2024a. Backtranslate what you are saying and i will tell who you are. *Expert Systems*, n/a(n/a):e13568.

Marco Siino, Maurizio Tesconi, and Ilenia Tinnirello. 2023. Profiling cryptocurrency influencers with few-shot learning using data augmentation and ELECTRA. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2772–2781. CEUR-WS.org.

Marco Siino and Ilenia Tinnirello. 2023. Xlnet with data augmentation to profile cryptocurrency influencers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2763–2771. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2022. T100: A modern classic ensemble to profile irony and stereotype spreaders. In *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, volume 3180 of *CEUR Workshop Proceedings*, pages 2666–2674. CEUR-WS.org.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024b. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

José Medardo Tapia-Téllez and Hugo Jair Escalante. 2020. Data augmentation with transformers for text classification. In *Advances in Computational Intelligence*, pages 247–259, Cham. Springer International Publishing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12):7789–7802.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

0x.Yuan at SemEval-2024 Task 5: Enhancing Legal Argument Reasoning with Structured Prompts

Yu-An Lu
National Chupei High School
luyuam0@gmail.com

Hung-Yu Kao
National Cheng Kung University
hykao@mail.ncku.edu.tw

Abstract

The intersection of legal reasoning and Natural Language Processing (NLP) technologies, particularly Large Language Models (LLMs), offers groundbreaking potential for augmenting human capabilities in the legal domain. This paper presents our approach and findings from participating in SemEval-2024 Task 5, focusing on the effect of argument reasoning in civil procedures using legal reasoning prompts. We investigated the impact of structured legal reasoning methodologies, including TREACC, IRAC, IRAAC, and MIRAC, on guiding LLMs to analyze and evaluate legal arguments systematically. Our experimental setup involved crafting specific prompts based on these methodologies to instruct the LLM to dissect and scrutinize legal cases, aiming to discern the cogency of argumentative solutions within a zero-shot learning framework. The performance of our approach, as measured by F1 score and accuracy, demonstrated the efficacy of integrating structured legal reasoning into LLMs for legal analysis. The findings underscore the promise of LLMs, when equipped with legal reasoning prompts, in enhancing their ability to process and reason through complex legal texts, thus contributing to the broader application of AI in legal studies and practice.

1 Introduction

The process of reasoning in legal arguments is a crucial aspect of applying legal knowledge in real-world scenarios. Mastery of this skill enables individuals to effectively address legal issues and ascertain the legality of various cases. Recently, the field of Natural Language Processing (NLP) has seen significant advancements, leading to the growing trend of utilizing Large Language Models (LLMs) as tools to augment human capabilities. Given the extensive and often complex body of legal knowledge, which can be challenging and time-consuming for the average person to learn, LLMs present an opportunity to comprehend this information and offer valuable assistance.

In light of this, the organizers of SemEval-2024 Task 5 (Held and Habernal, 2024) have compiled a dataset from the domain of U.S. civil procedure. This dataset includes introductory materials on various cases, a set of questions, potential argumentative solutions, and labels indicating the accuracy of these solutions. This initiative provides a framework for evaluating the effectiveness of LLMs in the legal arena, thereby contributing to the development of more sophisticated and capable language processing tools for legal applications.

In this task, we explored legal reasoning prompts in Large Language Models (LLMs) (Burton, 2017). Our focus was on investigating their effectiveness in differentiating argumentative solutions in civil procedure cases. The results show that by guiding an LLM to think step-by-step like a lawyer, it significantly outperforms both Chain of Thought (CoT) (Wei et al., 2023) reasoning and direct output methods.

2 Background

2.1 Related Works

Large Language Model(LLM): LLM is a kind of machine learning model in Nature Language Processing(NLP), pre-trained on large scale of text and can generate text based on previous context (Naveed et al., 2023). Beside LLM's usage in general works such as ChatGPT, LLM had also show its impressive abilities several professional fields like finance, medical and legal(Kaddour et al., 2023), such as BloombergGPT (Wu et al., 2023), Med-PaLM (Singhal et al., 2023) and ChatLaw (Cui et al., 2023).

LLM in Legal Field: Legal defined rules of human community, helping to make order to our life. But legal field have lots of professional knowledge, making obstacles to common people. Lots of legal

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 385–390

LLM or related methods are developed to solve this problem, (Cui et al., 2023) and (Nguyen, 2023) had trained the LLM in the legal field in Chinese justice, (Savelka et al., 2023) found that GPT-4 performed great in explaining legal concepts, (Savelka, 2023) found that some LLM already has legal knowledge in itself. These findings demonstrate the ability and potential of LLM to address legal-related issues.

Legal Reasoning: Legal reasoning is a kind of reasoning approach which had been used in law school teaching (Bentley, 1994), this approach initially aims to help law school students thinking legal questions in professional structure. (Burton, 2017) had make a overview of several legal reasoning approaches, such as 'CLEO' (Claim, law, evaluation, outcome). These approaches originally only used in legal field, until (Savelka, 2023) used these approaches as prompt in LLM, and found that these approach can make LLM's perform well on legal reasoning task, inspired by their works, we will try to use these approach flexible in LLM to help check the truthiness of argument reasoning in civil procedure.

2.2 Dataset Description

The dataset, developed for SemEval-2024 Task 5, focuses on the domain of U.S. civil procedure, aiming to test legal language models on their argument reasoning capabilities. It is meticulously structured to include a variety of components such as a brief introduction to each case, specific legal inquiries, proposed arguments, and in-depth analyses, making it a comprehensive tool for evaluating the nuanced understanding of legal texts. Each record within the dataset is uniquely identified and contains fields that detail the legal question at hand, a potential answer, and an indicator of the answer's accuracy (limited to the training and development subsets). Additionally, the dataset offers rich analyses, including both a focused excerpt relevant to the given answer and a complete solution explanation, along with supplementary explanations to contextualize the question further. Below Tab1 is an example of the dataset:

Attribute	Value
id	0
question	1. Redistricting. Dziezek, who resides in the Southern District of Indiana, sues Torruella...
answer	Case Study: Dziezek vs. Torruella and Hopkins
label	0
analysis	So the remaining question is whether the Western District of Kentucky, where Torruella resides, is a proper venue...
complete analysis	DLet's see. Under §1391(b)(1), venue is proper in a district where all defendants reside. But here they don't all reside in the same district...
explanation	Venue in most federal actions is governed by 28 U.S.C. §1391(b), which provides: (b) Venue in...

Table 1: Dataset example

3 Methodology

3.1 Legal Reasoning Prompts

Upon examining the dataset, we found that a significant portion of the legal knowledge pertinent to the argumentation is encapsulated within the 'Introduction' segment of the dataset. This observation suggests that the primary function of Legal Language Models (LLMs) is to facilitate reasoning from the provided text, as opposed to generating novel legal insights. Consequently, we have curated a selection of legal reasoning methodologies that adhere to the principle of meticulous analysis of the given text, progressively leading to a well-founded conclusion. The methodologies selected for this purpose are as follows:

- **TREACC** (Topic, Rule, Explanation, Analysis, Counterarguments, Conclusion): Provides a comprehensive analytical framework that includes discussions on counterarguments, aiding in the consideration and evaluation of all relevant aspects of a case in Legal Language Models (LLM).

- **IRAC** (Issue, Rule, Application, Conclusion): The fundamental structure for legal issue analysis, involving the identification of the issue, the rule, application of the rule to the facts, and drawing a conclusion.

- **IRAAC** (Issue, Rule, Application, Alternative Analysis, Conclusion): In addition to the basic steps of IRAC, this method incorporates an alternative analysis of the case, showcasing different facets of the issue.

- **MIRAC** (Material facts, Issues, Rules, Arguments, Conclusion): Emphasizes the importance of material facts and arguments by discussing them in detail before proceeding with the analysis and conclusion.

3.2 Experiments

In the devised architecture, attributes such as "question," "answer," and "explanation" were judiciously chosen to elicit from the Language Model (LLM) analyses predicated on legal reasoning, utilizing a zero-shot approach. A prompt was meticulously crafted, articulating the elements of legal reasoning methodologies, thereby casting the LLM in the role of a domain specialist tasked with the meticulous evaluation of responses in accordance with legal statutes. Moreover, the LLM was directed to encapsulate facets of the legal reasoning process within designated tags, e.g., <Topic> and </Topic>, to forestall omissions and diminish the likelihood of inaccuracies. Detailed elaboration of these prompts can be found in the appendixA for consultation.

Subsequent to the generation of analysis, these analyses were employed to instruct the LLM to adjudicate the cogency of the answers provided. The Mixtral-8x7B (Jiang et al., 2024) model, noted for its cost-efficiency and superior performance, was selected for our experimental evaluations.

4 Results

4.1 Official Evaluation Metrics

Agent	F1	Acc
TREACC	0.59	0.62
IRAC	**0.60**	**0.63**
IRAAC	**0.60**	**0.63**
MIRAC	**0.60**	**0.63**
CoT	0.53	0.58
Directly	0.49	0.55

Table 2: Performance of different methods.

As demonstrated in Table 2, various methods exhibit distinct performances on the test dataset. In contrast, methodologies such as CoT and Direct Output solely leverage "question," "answer," and "explanation" to prompt the LLM to discern the veracity of the answers.

Our analysis revealed that strategies incorporating legal reasoning prompts uniformly outperformed the CoT and Direct Output approaches, underscoring the efficacy of our methodology. Intriguingly, the IRAC, IRAAC, and MIRAC methods manifested identical performance metrics on the test dataset. A deeper examination of the prediction outcomes suggested that this phenomenon could be attributed to the pronounced similarity in the analyses engendered by these methods.

The analyses generated predominantly adhered to the stipulated procedural steps, fostering a methodical and layered approach to thinking and reasoning. With the exception of approximately 1% of instances, the data conformed to our prompts, yielding comprehensive structural outputs. An illustrative excerpt from an analysis employing the IRAC method is presented below:

> <issue>...The pivotal legal question concerns the appropriate venue within the Southern District of New York for a negligence lawsuit...</issue> <rule>...Venue is determined by 28 U.S.C. §1391(b)(3), which stipulates its appropriateness in any jurisdiction where a defendant resides, provided all defendants are state residents...</rule> <application>...In initiating a negligence suit against Lenoir and Sorrel in Canada, with Sorrel being served in New York, the query arises regarding the venue's suitability for both defendants...</application> <conclusion>...Legal ambiguities surrounding the residency requirement for defendants complicate the determination of an eligible venue for proceeding with the case...</conclusion>

Our team, identified as "Yuan_Lu" on the leaderboard, achieved a commendable 10th place among 21 participating teams.

4.2 Case Discussion

Here's a comparison of a cases that classify by **IRAC**, **CoT** and **Directly** output methods.

- *Question*: Black White Taxicab, incorporated in Kentucky, seeks to do business at the Bowling Green railroad station but faces opposition from Brown Yellow Taxicab's exclusive contract with the railroad. Given Kentucky courts' stance against such contracts as against public policy, Black White sues Brown Yellow to prevent it from enforcing the contract. Assuming diversity jurisdiction and the amount-in-controversy requirement are met, should the

federal court apply Kentucky law under the Rules of Decision Act?

- *Answer*: Apply Kentucky law under the Rules of Decision Act.

- *Explanation*: This question references the historical case of Black White Taxicab Co. v. Brown Yellow Taxicab Co., highlighting the manipulation of corporate structure to create diversity jurisdiction and bring a case to federal court where federal common law, under Swift v. Tyson, would allow enforcement of a contract deemed invalid under state law. This scenario underscores the tension between federal and state law in diversity cases prior to the Erie decision.

- *Label*: **False**

4.2.1 Directly output's answer

> *The answer is TRUE. This interpretation relies on the Rules of Decision Act as understood before the Erie decision, suggesting that the federal court would apply state law unless a federal statute, constitutional provision, or treaty dictates otherwise.*

The direct output mistakenly considers the pre-Erie understanding applicable, ignoring the significant shift in legal practice introduced by Erie Railroad Co. v. Tompkins.

4.2.2 CoT's answer

> *Answer: TRUE, with a caveat. The response suggests a nuanced approach where federal courts, under the pre-Erie doctrine of Swift v. Tyson, might not strictly apply state law but would consider it alongside a broader interpretation of common law principles, particularly regarding public policy against exclusive contracts for taxi services at railroad stations.*

CoT's analysis presents a sophisticated understanding but incorrectly aligns with the pre-Erie legal framework, not recognizing Erie's directive for federal courts to adhere strictly to state law in diversity cases.

4.2.3 IRAC's answer

> *The analysis is NOT true. Erie Railroad Co. v. Tompkins overruled the Swift v. Tyson doctrine, mandating that federal courts apply state law in diversity cases. This correct interpretation implies that, in the scenario provided, the federal court would indeed apply Kentucky law, reflecting the post-Erie legal landscape accurately.*

The IRAC method accurately reflects the current legal standard, demonstrating a correct understanding of the Erie doctrine's requirement for federal courts to apply state law in diversity cases, thus providing the most legally sound interpretation.

4.2.4 Summary

Both Direct Output and CoT inaccurately conclude based on outdated pre-Erie interpretations, with Direct Output oversimplifying and CoT inadequately applying post-Erie legal standards. In contrast, the IRAC method accurately applies the Erie doctrine, demonstrating a nuanced understanding of current legal principles by methodically breaking down the issue and applying the correct rule. This approach not only ensures precision in legal analysis but also aligns conclusions with contemporary legal frameworks, showcasing its distinct contribution to legal reasoning and highlighting the importance of structured analysis in achieving accurate legal interpretations.

5 Conclusion

This study embarked on an exploration of the synergy between Large Language Models (LLMs) and legal reasoning methodologies to enhance the processing and understanding of legal texts. By integrating structured legal reasoning prompts derived from methodologies such as TREACC, IRAC, IRAAC, and MIRAC into the framework of LLMs, we demonstrated the potential of this approach to improve the models' capacity for legal argument evaluation.

References

Duncan Bentley. 1994. Using structures to teach legal reasoning. *Legal Education Review*, 5(2).

Kelley Burton. 2017. "think like a lawyer" using a legal reasoning grid and criterion-referenced assessment rubric on irac (issue, rule, application, conclusion). *Journal of Learning Design*, 10:57–68.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases.

Lena Held and Ivan Habernal. 2024. SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models.

Ha-Thanh Nguyen. 2023. A brief report on lawgpt 1.0: A virtual legal assistant based on gpt-3.

Jaromir Savelka. 2023. Unlocking practical applications in legal domain: Evaluation of gpt for zero-shot semantic annotation of legal texts. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, page 447–451, New York, NY, USA. Association for Computing Machinery.

Jaromir Savelka, Kevin D. Ashley, Morgan A. Gray, Hannes Westermann, and Huihui Xu. 2023. Explaining legal concepts with augmented large language models (gpt-4).

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance.

A Prompts of Legal Reasoning Prompts

A.1 TREACC

''' Question: question Answer: answer Explanation: explanation

Analyze the given legal case scenario following these structured steps:

<topic> Identify and briefly describe the main legal issue. </topic> <rule> State the relevant legal principles or statutes that apply to the legal issue identified. </rule> <explanation> Provide a detailed explanation of the legal principles or statutes, including their background, scope, and examples of their application in previous cases. </explanation> <analysis> Apply the facts of the case to the legal principles or statutes, and evaluate how these facts fit or support the rules. </analysis> <counterarguments> Identify and explain any potential counterarguments or opposing views to the main analysis. </counterarguments> <conclusion> Summarize the analysis and provide a clear conclusion or opinion on the main legal issue. </conclusion> Use the given data to perform a structured analysis and present your findings under each labeled section. Don't forget to add label to each part, Once you're sure all tags have been added, say "I'm sure I've added all tags" at the end. Streamline the length. '''

A.2 IRAC

''' Question: question Answer: answer Explanation: explanation

Analyze the given legal case scenario following these structured steps:

<issue>Identify the key legal issue at the heart of the scenario.</issue> <rule>Detail the specific laws or legal principles that govern the identified issue.</rule> <application>Examine how the laws or principles apply to the facts of the case, discussing the legal merits of the case based on this application.</application> <conclusion>Conclude by synthesizing the analysis to state the likely outcome of the case based on the application of the rule to the issue.</conclusion>

Use the given data to perform a structured analysis and present your findings under each labeled section. Don't forget to add label to each part, Once you're sure all tags have been added, say "I'm sure

I've added all tags" at the end. Streamline the length. "'

A.3 IRAAC

"' Question: question Answer: answer Explanation: explanation

Analyze the given legal case scenario following these structured steps:

<issue>Identify the central legal issue present in the case.</issue> <rule>Articulate the rule of law that applies to the issue, including any relevant legal standards or precedents.</rule> <application>Analyze how the rule of law should be applied to the particular facts of the case, considering all relevant factors.</application> <alternative_analysis>Discuss an alternative legal analysis or perspective that might lead to a different outcome, considering other possible interpretations of the law or facts.</alternative_analysis> <conclusion>Provide a final conclusion that takes into account both the primary and alternative analyses, and state the most persuasive legal position.</conclusion>

Use the given data to perform a structured analysis and present your findings under each labeled section. Don't forget to add label to each part, Once you're sure all tags have been added, say "I'm sure I've added all tags" at the end. Streamline the length. "'

A.4 MIRAC

"' Question: question Answer: answer Explanation: explanation

Analyze the given legal case scenario following these structured steps: <material_facts>Begin by presenting the material facts of the case, focusing on those critical to the legal issues.</material_facts> <issues>Identify the specific legal issues that arise from these material facts.</issues> <rules>State the legal rules and principles that will be used to address these issues.</rules> <arguments>Develop arguments that apply the legal rules to the issues, considering the material facts and any relevant legal arguments, including policy considerations where applicable.</arguments> <conclusion>Conclude with a summary that encapsulates the findings from the application of the rules to the issues, supported by the arguments, and clearly state the resolved position on the case.</conclusion>

Use the given data to perform a structured analysis and present your findings under each labeled section. Don't forget to add label to each part, Once you're sure all tags have been added, say "I'm sure I've added all tags" at the end. Streamline the length. "'

Groningen team D at SemEval-2024 Task 8: Exploring data generation and a combined model for fine-tuning LLMs for Multidomain Machine-Generated Text Detection

Thijs Brekhof, Xuanyi Liu, Joris Ruitenbeek, Niels Top, Yuwen Zhou
University of Groningen
t.j.brekhof@student.rug.nl, x.liu.69@student.rug.nl,
j.e.j.ruitenbeek@student.rug.nl, n.top@student.rug.nl, y.zhou.74@student.rug.nl

Abstract

In this system description, we report our process and the systems that we created for the subtasks A monolingual, A multilingual, and B for the SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection. (Wang et al., 2024) This shared task aims at discriminating between machine-generated text and human-written text. Subtask A focuses on detecting if a text is machine-generated or human-written both in a monolingual and a multilingual setting. Subtask B also focuses on detecting if a text is human-written or machine-generated, though it takes it one step further by also requiring the detection of the correct language model used for generating the text. For the monolingual aspects of this task, our approach is centered around fine-tuning a deberta-v3-large LM. For the multilingual setting, we created a combined model utilizing different monolingual models and a language identification tool to classify each text. We also experiment with the generation of extra training data. Our results show that the generation of extra data aids our models and leads to an increase in accuracy.

1 Introduction

The SemEval-2024 shared task focuses on multigenerator, multidomain, and multilingual black-box machine-generated text detection. The shared task is split into three different subtasks. Each subtask is monolingual except for the first subtask, which has a monolingual (English) and a multilingual track. The languages covered in this shared task include English, Chinese, Russian, Urdu, Indonesian, Italian, German, and Arabic.

This paper presents our the systems that we created for the shared task. The paper provides an overview of our research strategies and results for subtasks A and B.

Subtask A focuses on the detection of machine- versus human-written text, we differentiate between mono- and multilingual data. Our approach involves fine-tuning LLMs, DeBERTa-v3 (large) in particular. We experimented with different parameters for the model, searching for the best performance possible.

Subtask B extends the challenge presented in subtask A, we now attempt to recognize the specific language model used for text generation. We do this in addition to distinguishing between human and machine-generated text. We again use DeBERTa-v3 (large) to classify the data. To optimize model accuracy, we fine-tune hyperparameters.

Additionally, we generate extra Wikipedia articles to further expand the training data. We hypothesize that extra data will lead to better model performance, and thus better applicability to real-world applications. Our research focuses on finding both the best possible language model settings to recognize machine- and human-written text and distinguish between different language generation models. Our code and the additionally generated data can be found on Github[1]

2 Related work

Previous research has been done on the topic of automatically discriminating between human-text and machine-generated text (Chichirau et al., 2023), where DeBERTa (v3) (He et al., 2021) is utilized as a target-only classifier. The model can distinguish machine translations well when tested on the test set after training on texts generated from different source languages and different ma-

[1] https://github.com/thijsbrekhof1/RUG-D-at-SemEval2024-task8

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 391–398

	Train	Dev	Test
Subtask A-Mono	119757	5000	34272
Subtask A-Multi	172417	4000	42378
en	136589	0	28200
ar	0	1000	2103
ru	0	2000	0
zh	11934	0	0
id	5995	0	0
ur	5899	0	0
bg	12000	0	0
de	0	1000	6000
it	0	0	6075
Subtask B	71027	3000	18000

Table 1: Statistics of train, dev, and test sets provided by organizers

chine translation systems. They found that both the monolingual and multilingual DeBERTa models outperformed other LLMs that they evaluated.

Langid.py (Lui and Baldwin, 2012) is a supervised language identification tool trained using a naive Bayes classifier. Langid.py has the following advantages: fast, usable off-the-shelf, unaffected by domain-specific features (e.g. HTML, XML, markdown), single file with minimal dependencies, and flexible interface. Langid.py was applied in our system to identify the multi-language training set of subtask A and we found that it can identify languages with very high accuracy.

3 Data

The dataset provided by the shared task creators originates from the benchmark M4 (Wang et al., 2023). M4 is a comprehensive dataset encompassing machine-generated text from diverse generators, domains, and languages. M4 focuses on the development of automated systems for detecting machine-generated text and identifying potential abuse.

The dataset comprises text samples sourced from various platforms, including Wikipedia, Reddit, WikiHow, PeerRead, Arxiv, Chinese QA, Urdu News, Russian RuATD, Indonesian News, and Arabic Wikipedia. It spans multiple languages and domains, presenting a rich and diverse collection of machine-generated text for analysis and classification.

Table 1 presents the statistics of the dataset, including the number of samples in the train, dev, and test sets for subtasks A and B. For subtask A, both monolingual (subtask A-Mono) and multilingual (subtask A-Multi) tracks are included, with train, dev, and test set sizes specified for each language. Subtask B involves multi-way classification of machine-generated text and includes corresponding train, dev, and test set sizes.

4 System overview

This section presents an overview of the methods we employed for subtask A, both the monolingual and multilingual data setting, as well as subtask B. We follow previous work on a similar topic (Chichirau et al., 2023), by fine-tuning LLMs, predominantly DeBERTa (He et al., 2021), on this task. We were further stimulated to explore this model specifically, as DeBERTa is developed as an improvement over the RoBERTa model (Liu et al., 2019), the latter being employed by the task organizers as a baseline. Specifically, we looked at using both the base and large variants of deberta-v3, as this improved version of DeBERTa is reported to significantly outperform previous iterations on numerous tasks.

As the goal of this task was to create systems that can discriminate between human-written and machine-generated text regardless of generator, textual domain, or language, we opted not to preprocess our data any further than what the task organizers already did. This will keep our data as close to instances that can be encountered in real-world scenarios as possible. We fine-tuned these pre-trained language models using the Transformers library from Huggingface (Wolf et al., 2020).

4.1 Subtask A: Monolingual

For the monolingual track of subtask A, we evaluated the performance of the base (86M parameters) and large (304M parameters) variants of DeBERTa. We tested out numerous combinations of hyperparameters such as learning rate, batch size, maximum input sequence length, and epochs to found out which model would perform best. The large DeBERTa model emerged superior over the base model, ostensibly due to its larger model size.

For this track of the task, we also experimented with generating additional training data. The goal for this subtask is for our model to differentiate between human-written and machine-generated text, regardless of what generative model was used to obtain data. We were inclined to experiment with

additional data generation by a model different from the ones already present in the provided base dataset, as this should allow our model to generalize better across generators and not learn only about those present in the base dataset. For potential real-world applications, this would be especially interesting to experiment with, as in such scenarios there would be no prior indication of what model could be used to generate such texts.

We employed Llama 2 (Touvron et al., 2023) to generate additional articles in the style of Wikipedia and manually skimmed through the generated texts to see if they were on a comparable level to the data provided by the task organizer. Subsequently, we took the hyperparameter configurations of our best-performing model trained on only base data and trained a new model using the same configurations on a combination of the base data and our additionally generated articles. The selection of Wikipedia as our domain of focus is based on its comprehensible documentation and the strong performance demonstrated by LLama 2 in generating texts within this specific domain.

4.2 Subtask A: Multilingual

Different from the monolingual strategy, we created a combined model for this subtrack. We explored a way to use separate monolingual models for different languages after determining the language of each text. After discovering that there was no data in the same language both in the original train and dev set (see Table 1, we decided to merge the two data sets and extract each language separately for analysis. We embarked on a language-specific modeling approach, recognizing the importance of selecting models optimized for each language's unique characteristics.

To determine the most suitable approach for each language, we compared the performance of multilingual DeBERTa with specific monolingual models. We employed a 10-fold cross-validation approach within each language, evaluating models based on accuracy and standard deviation. The best-performing model for each language was selected for further evaluation.

Upon completion of the cross-validation procedure, we selected the model that exhibited the highest performance on the development set for each language. The selected models were then applied to the test set for final evaluation, encompassing the full spectrum of languages represented in the dataset. To handle the multilingual nature of the test set effectively, we employed the language identification tool Langid, to discern the language of each text sample, which enabled us to tailor model predictions to the specific linguistic context of each sample.

Notably, we also employed Llama 2 to generate additional training data for each language. We utilized a 10-fold cross-validation process to assess the impact of additional training data on model performance across different languages and only kept those that improved the results.

4.2.1 LangID

In our multilingual subtask A experiment, we proposed the idea of using specific language models per language instead of a single model for each of the languages. Our motivation was that this approach could improve the accuracy of discriminating between machine-generated text and human-written texts better than a single multilingual model could. To achieve this goal, we employed LangID to enable language-specific modeling. After merging the train and dev sets and extracting samples for each language separately, we utilized LangID to determine the language of each text sample in the test set and employed MDeBERTa-v3-base for languages that were not in the train or dev sets and could not be recognized by LangID. Thus, we were able to effectively handle the multilingual nature of the task.

4.3 Subtask B

For this subtask, we, similarly to our approach for subtask A, compared the performance of the base and large variant of DeBERTa. By testing out different values for epochs, learning rate, maximum input sequence length, and batch size, we obtained the hyperparameter configurations of our best-performing model. The large variant of DeBERTa once again outperformed the base version.

We opted not to use additionally generated data for this subtask. The goal of subtask B is to determine not only if the text is human-written or machine-generated but also what generative model was used to do so. This would make generating data by models outside of the already provided list of models in the base dataset futile.

4.4 Generating data

While we realize that it is not allowed to add additional data for the shared task we see generating

it as a real-world contribution that can also easily be done by others. We generated our own extra training data with the use of Llama 2 (Touvron et al., 2023). Starting off, we wanted to exploit the largest model available, because this should offer the best performance in data generation. However, due to limited resources, we opted to utilize the 7 billion parameter version.

We focussed our generation endeavors on languages that were already in the dataset but were highly underrepresented. These included Russian, Arabic, German, and Indonesian. For each of these languages, we extended the dataset so that each of these languages had a total of 30,000 samples. Notably, for every sample generated by the model, we also included a human-written counterpart in the dataset. By doing this, we aimed to maintain a balance between computer- and human-written data in the training and development sets.

To match the already generated Wikipedia articles in the dataset, we adopted a similar method to the original M4 dataset, as outlined by (Wang et al., 2023). Using the Wikipedia dataset available on HuggingFace (Wikimedia-Foundation, 2023), we randomly selected articles with a minimum length of 1,000 characters. Subsequently, we prompted Llama 2 to generate Wikipedia articles based on provided titles. As an extra criterion, we told the model that the resulting articles should contain at least 250 words, as this was also the criteria used in the original paper (Wang et al., 2023). This approach enabled us to enrich our dataset across multiple languages, with the purpose of increasing the performance of our models.

5 Experimental setup

5.1 Datasets and Evaluation Metrics

For both subtask A's monolingual part and subtask B, we utilized standard data splits: train, dev, and test sets. The train set was employed for model training, the dev set for monitoring performance and hyperparameter tuning, and the test set for final evaluation. Accuracy is the main evaluation metric to assess model performance in each task.

For multilingual subtask A, we adopted a different strategy, as motivated in Section 4.2. We concatenated the train and dev sets, extracted samples for different languages, and employed separate models for each language. We utilized the 10-fold cross-validation approach within each language to select the most suitable model based on accuracy and standard deviation. The selected models from each language were then used to predict the test set.

5.2 Training Details

For monolingual subtask A, the final selected hyperparameters were as follows: batch size 2, gradient accumulation 64, learning rate 1e-5, three epochs, formatting style fp16, and an input length of 1024 tokens.

For multilingual subtask A, we employed uniform hyperparameters throughout the 10-fold cross-validation process within each language. These hyperparameters included a learning rate of 2e-5, three epochs, a formatting style of fp16, and an input length of 512 tokens.

For subtask B, the following hyperparameters were identified as optimal: batch size 4, gradient accumulation 32, learning rate 1e-5, three epochs, formatting style fp16, and an input length of 512 tokens.

All of our hyperparameter values were chosen after extensive experimentation on the dev set to optimize model performance. A full list of all the hyperparameter values that we experimented with regarding the monolingual subtasks can be found by referring to Appendix A. Regarding multilingual subtask A, specific model selection and results for each language can be seen in Table 6 of Appendix B.

Additionally, all training processes were conducted on several Nvidia A100 and V100 GPUs.

6 Results and Analysis

In this section, we show and analyze the results achieved for each of the subtasks. Table 2 shows the quantitative results we achieved when running our models on our dev set and the organizer's test set. Tables 4, 7 and 5 in the appendix show the accuracy across languages and the impact of the usage of extra data on each subtask. Besides that, we made a qualitative analysis to find out where we think our systems make the most mistakes.

6.1 Analysis

Our analysis showed us several noteworthy points. First, our monolingual models achieved significantly higher scores on the dev sets than on the test set, as can be seen in Table 2. A reason for this could be the introduction of texts created by LLMs

Subtask	Baseline	Dev	Test
A Monolingual	88.46%	87.80%	63.68%
A Multilingual	80.88%	65.90%	71.79%
B	74.60%	72.80%	61.50%

Table 2: Scores of each subtask in dev and test compared to the baseline.

that our system had not seen before. This shows us a risk our systems may lack robustness against different types of LLMs. Our multilingual system did perform better on test than on dev, however, which could be related to the different ratios of languages present in both datasets. e.g., more German texts were present in the test dataset than in the dev dataset, and our system is able to classify them effectively, which can be seen in Table 6.

Furthermore, our systems were unable to effectively detect human-written texts, in both the mono- and multilingual tasks, when classifying the test set. In subtask A monolingual, our system was able to get very impressive scores on all texts created by generative models, though it had a lackluster performance on human-written text. This might indicate our model's inclination to classify a text as machine-generated over human-written. Subtask B has a very similar distribution of predictions, the only notable exception being the obstacle of detecting texts written by Cohere.

Also noticeable was the performance of texts generated by the Llama 2 model. Both our models with- and without added data scored badly on these texts. What is interesting, is that the extra data added by us originates from Llama 2. A reason for this could be that we used the smaller, 7 billion parameter, version of Llama 2 due to performance and runtime issues.

We can see that both in the mono- and multilingual data setting of subtask A our model's performance had improved after training on our extra generated data. Although the increase in accuracy of the monolingual model was negligible, the multilingual model had a notable improvement in score. We propose that this stems from the absence of certain languages in the training set, which we were able to supplement with our extra data. Because of this, the monolingual models we employed in the multilingual setting were able to perform better.

7 Discussion/Conclusion

In conclusion, we think our participation in the shared task resulted in some valuable insights into the challenges of machine- versus human-written text. Despite our efforts, our systems unfortunately fell short of surpassing the baseline scores established by the task organizers.

Across the different subtasks, our models showed varying performance. For subtask A monolingual, our models achieved some promising results on the development set, with an accuracy of 87.80%. However, our model did not manage to generalize enough, leading to an accuracy of 63.68% on the test set.

For the multilingual part of subtask A, our model reached 65.90% on the development set. In this case, the model did manage to generalize the data, leading to an accuracy of 71.79% on the test set. However, this was still below our expectations, and the baseline accuracy of 80.88%.

In subtask B, our models struggled to identify the specific language model used for text generation accurately, with accuracies of 72.80% on the development, and 61.50% on the test set. Despite optimizing hyperparameters and training on both original and additional data, our models failed to outperform the baseline accuracy of 74.60%.

We think our analysis revealed several points for improvement. Our models tended to misclassify human-written text, indicating a potential bias towards machine-generated content. Furthermore, the models seemed unable to generalize, leading to worse performance on the test set for monolingual task A.

Moving forward, we think there are many improvements to be made. Future research could focus on using other model architectures or exploring other data augmentation techniques. Also, training the model in more languages could improve the performance of multilingual models. Of course, using larger pre-trained models could also lead to an easy increase in performance, although it does require significant resources. Lastly, our findings also show generating extra training data is essential for improving model performance. Therefore, a promising direction for future work is to explore new data sources and methods to create richer and higher-quality training data to further improve the performance and generalization ability of the model.

8 Acknowledgements

This submission has been carried out as part of the 2023-2024 edition of the master course Shared Task Information Science (LIX026M05) at the University of Groningen, taught by Lukas Edman and Antonio Toral. We want to express our gratitude to the university and to both of our lecturers for assisting us in this project.

References

[Abdaoui et al.2020] Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of mutlilingual bert. In *SustaiNLP / EMNLP*.

[Antoun et al.2020] Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

[Chan et al.2020] Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

[Chichirau et al.2023] Malina Chichirau, Rik van Noord, and Antonio Toral. 2023. Automatic discrimination of human and neural machine translation in multilingual scenarios. In Mary Nurminen, Judith Brenner, Maarit Koponen, Sirkku Latomaa, Mikhail Mikhailov, Frederike Schierl, Tharindu Ranasinghe, Eva Vanmassenhove, Sergi Alvarez Vidal, Nora Aranberri, Mara Nunziatini, Carla Parra Escartín, Mikel Forcada, Maja Popovic, Carolina Scarton, and Helena Moniz, editors, *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 217–226, Tampere, Finland, June. European Association for Machine Translation.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

[Guo et al.2023] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arxiv:2301.07597*.

[He et al.2021] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

[Kuratov and Arkhipov2019] Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language.

[Liu et al.2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

[Lui and Baldwin2012] Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30.

[Touvron et al.2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

[Wang et al.2023] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.

[Wang et al.2024] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. SemEval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico, June.

[Wikimedia-Foundation2023] Wikimedia-Foundation. 2023. Wikimedia downloads.

[Wolf et al.2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf,

Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Scao, Sylvain Gugger, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. pages 38–45, 01.

A Hyperparameters

Hyperparameter	Value
Learning rate	1e-5, 2e-5, 5e-5, 5e-6
Batch size	16, 32, 64, 128
Epoch	1, 2, 3
Input length	512, 768, 1024

Table 3: The full hyperparameter search space employed for our DeBERTa-v3-large model in both subtask A monolingual and subtask B.

B Detailed scores

Data	Model	Accuracy
A Monolingual - Original Data		
	Overall	63.61% (± 2.60E-3)
	Human	23.56% (± 3.33E-3)
	GPT4	99.77% (± 8.81E-4)
	Cohere	100.0% (0)
	ChatGPT	100.0% (0)
	Bloomz	99.1% (± 1.72E-3)
	Dolly	100.0% (0)
	Davinci	99.97% (± 3.33E-4)
A Monolingual - Added Data		
	Overall	63.68% (± 2.60E-03)
	Human	24.23% (± 3.36E-03)
	GPT4	99.9% (± 5.77E-04)
	Cohere	100.0% (0)
	ChatGPT	100.0% (0)
	Bloomz	96.2% (± 3.49E-03)
	Dolly	100.0% (0)
	Davinci	100.0% (0)

Table 4: Accuracy scores on the test set for subtask A Monolingual with original and added data.

Model	Accuracy
Overall	61.54% (± 3.63E-03)
Human	13.53% (± 1.37E-03)
Bloomz	99.43% (± 6.25E-03)
Dolly	86.1% (± 6.32E-03)
ChatGPT	99.93% (± 4.71E-04)
Cohere	1.23% (± 2.02E-03)
Davinci	69.0% (± 8.44E-03)

Table 5: Accuracy scores for subtask B on the test set.

Lang.	Model	Accuracy	Reference
en	deberta-v3-base	**95.9% ± 5.82E-3**	(He et al., 2021)
en	mdeberta-v3-base	95.6% ± 5.94E-3	(He et al., 2021)
ar	bert-base-arabert	**94.0% ± 4.10E-2**	(Antoun et al., 2020)
ar	mdeberta-v3-base	90.5% ± 4.46E-2	(He et al., 2021)
ru	rubert-base-cased	98.7% ± 8.12E-3	(Kuratov and Arkhipov, 2019)
ru	mdeberta-v3-base	**98.8% ± 6.00E-3**	(He et al., 2021)
zh	chatgpt-detector-roberta-chinese	**97.6% ± 5.64E-3**	(Guo et al., 2023)
zh	bert-base-chinese	96.8% ± 1.04E-2	(Devlin et al., 2019)
zh	mdeberta-v3-base	96.9% ± 1.37E-2	(He et al., 2021)
id	bert-base-indonesian-522M	**99.4% ± 4.68E-3**	
id	mdeberta-v3-base	98.8% ± 7.85E-3	(He et al., 2021)
ur	mdeberta-v3-base	**99.98% ± 5.08E-4**	(He et al., 2021)
bg	bert-base-en-bg-cased	97.2% ± 6.29E-3	(Abdaoui et al., 2020)
bg	mdeberta-v3-base	**99.3% ± 4.99E-3**	(He et al., 2021)
de	bert-base-german-cased	92.9% ± 3.53E-2	(Chan et al., 2020)
de	gbert-base	**93.8% ± 1.99E-2**	(Chan et al., 2020)
de	mdeberta-v3-base	91.0% ± 4.4E-2	(He et al., 2021)

Table 6: The accuracy and standard deviation of different models in each language under 10-fold cross validation. The best-performing models (in bold) were utilized in our combined model for multilingual subtask A. We only employed one (multilingual) model for Urdu, as we could not find any monolingual models trained on that language.

Data	Model	Accuracy	Language	Accuracy
A Multilingual - Original data				
	Overall	70.11% (± 2.22E-03)	English	72.32% (± 2.66E-03)
	Human	40.89% (± 4.28E-03)	German	84.45% (± 4.68E-03)
	ChatGPT	83.91% (± 3.51E-03)	Arabic	57.73% (± 1.08E-02)
	Bloomz	100.0% (0)	Italian	50.01% (± 6.42E-03)
	Davinci	99.9% (± 5.77E-04)		
	Llama 2	50.01% (± 6.42E-03)		
	Dolly	99.93% (± 4.71E-04)		
	Cohere	100.0% (0)		
	Jais-30b	61.29% (± 3.91E-02)		
A Multilingual - Added data				
	Overall	71.79% (± 2.19E-03)	English	72.32% (± 2.66E-03)
	Human	40.89% (± 4.28E-03)	German	90.92% (± 3.71E-03)
	ChatGPT	90.14% (± 2.85E-03)	Arabic	73.13% (± 9.67E-03)
	Bloomz	100.0% (0)	Italian	50.01% (± 6.42E-03)
	Davinci	99.9% (± 5.77E-04)		
	Llama 2	50.01% (± 6.42E-03)		
	Dolly	99.97% (± 3.33E-04)		
	Cohere	100.0% (0)		
	Jais-30b	80.0% (± 3.21E-02)		

Table 7: Accuracy scores and language-specific accuracies on the test set for subtask A Multilingual with original and added data.

Kathlalu at SemEval-2024 Task 8: A Comparative Analysis of Binary Classification Methods for Distinguishing Between Human and Machine-generated Text

Lujia Cao and **Ece Lara Kılıç** and **Katharina Will**
University of Tübingen
{lujia.cao, ece-lara.kilic, katharina.will}@student.uni-tuebingen.de

Abstract

This paper investigates two methods for constructing a binary classifier to distinguish between human-generated and machine-generated text. The main emphasis is on a straightforward approach based on Zipf's law, which, despite its simplicity, achieves a moderate level of performance. Additionally, the paper briefly discusses experimentation with the utilization of unigram word counts.

1 Introduction

This paper addresses the task of classifying textual data as human or machine-generated, focusing on Subtask A Wang et al. (2024) with monolingual English data. The rise of technologies like ChatGPT has led to a surge in the use of machine-generated content in academia and workplaces. The task is crucial for ensuring the authenticity of texts, especially as individuals may potentially claim authorship of machine-generated content as their own work, raising concerns about academic integrity. By focusing on Subtask A and utilizing English-language data, this research addresses the challenges associated with the increasing prevalence of machine-generated text in academic and professional contexts, offering effective classification methods. In this paper, we use simple methods based on linguistic intuition to distinguish between human and machine-generated text. Our primary approaches involve leveraging Zipf's Law as one method, and employing word unigram counts as another.

We explored multiple approaches, ultimately narrowing our focus to two strategies. Although we submitted only one approach for leaderboard consideration, we believe the other one offers valuable insights as well. Surprisingly, we found that our simple methods based on linguistic intuition can rival the performance of large language models in the same task. While this approach could have been applied to the multilingual track, regrettably, time constraints prevented us from pursuing this direction. We ranked in the middle compared to other teams participating in this task.

2 Background

In configuring our task, we utilized the subtask A monolingual training data to train all three approaches, encompassing texts from both humans and models such as ChatGPT OpenAI (2022), Cohere Cohere (n.d.), Davinci OpenAI (n.d.), and Dolly Hugging Face (n.d.). Each data entry included the text, its source, the model used, the assigned label (either 0 or 1), and a unique identifier. For development purposes, we employed the subtask A monolingual development data, which featured texts generated by humans and the Bloomz BigScience (n.d.) model, along with their corresponding source, model, label, and ID. The final test data exclusively included texts and their respective IDs, with all other information omitted. The output data comprised jsonl files containing only the text IDs and their predicted labels (either 0 or 1). These files were generated once using the development data to refine our approach and again for the final test data, aligning with the task objective of predicting the label for a given text using our approach.

Our research delved into the practical implementation of Zipf's Law for binary classification. While consulting Linders and Louwerse's paper Linders and Louwerse (2020), as well as Nguyen-Son et al. Nguyen-Son et al. (2017), we found theoretical mentions of its potential application. However, none of these sources provided an actual approach. In contrast, our approach involves the concrete implementation of Zipf's Law, resulting in a functioning system.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 399–402

3 System Overview

3.1 Zipf's Law

Following an extensive review of the literature concerning methodologies aimed at discriminating between human and machine-generated textual content, our inquiry identified Zipf's Law as a potentially promising avenue of investigation. Despite the limited prevalence of existing methodologies leveraging this distribution, we deemed it worthy of investigation. Our rationale for pursuing this direction stems from the observed advantages in terms of computational efficiency and simplicity compared to Large Language Model (LLM) based approaches, which typically incur higher computational demands.

Zipf's Law is characterized by the following equation.

$$f = \frac{C}{r^s}$$

- $f(r)$ represents the frequency of the rank rth term.
- C is a constant.
- s is the Zipf exponent, typically close to 1.

This formula illustrates the inverse relationship between the frequency of a term and its rank in a given dataset, with the Zipf exponent governing the rate of decline in frequency as rank increases.

Our code initiates by tokenizing the text into individual words, followed by the computation of each word's frequency within the text. This preliminary step is pivotal for acquiring the empirical frequency distribution of words. After computing word frequencies and their corresponding ranks, we fit a curve to the Zipfian distribution. This step takes into account the Zipfian distribution function, word ranks, and frequencies as input parameters. By optimizing the scaling parameter s of the Zipfian distribution, fitting the observed data to a curve reveals the text's adherence to Zipf's law. This process aims to determine the optimal parameters (such as the scaling parameter s and constant C) for the Zipfian distribution function.

Leveraging the parameter s, the Zipfian distribution, we computed mean values for texts of label 0 and label 1. Subsequently, we determined their midpoint (-0.125) to serve as the threshold for classifying a text as either label 0 (human-generated) or label 1 (machine-generated).

In the system overview, following label prediction for the development data in subtask A monolingual track, we conduct a thorough evaluation by manually computing a preliminary F1 score using the actual labels as references. Our prediction process involves assigning labels based on Zipfian values, where values below the predefined threshold receive label 0 and those exceeding it are labeled as 1. We systematically apply the Zipfian distribution method to the texts extracted from the development data, facilitating precise label determination during subsequent analysis.

Our system obtained an F1 score of 0.72 on the development set. We used the same threshold/model to predict the labels on the test set.

The F1 score obtained on the official leaderboard for this approach yielded a value of 0.729. This metric provides a robust assessment of our approach's performance in the context of the shared task.

	label 0	label 1
min	-0.539	-1.778
max	2.212e-09	-1.550e-10
mean	-0.111	-0.139

Table 1: Zipfian distribution of labels 0 and 1

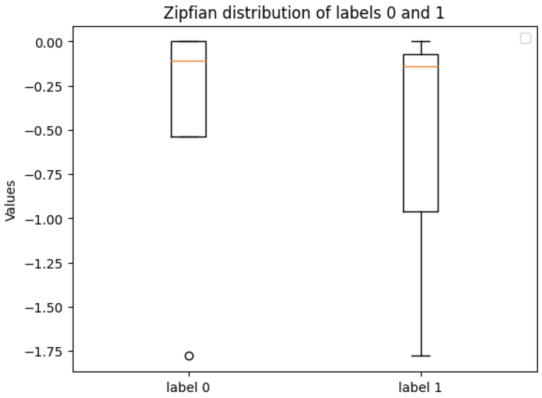


Figure 1: Zipfian distribution of labels 0 and 1

3.2 Unigram

Although we did not submit the predictions of the unigram approach, we will clarify its setup here. This method mirrors the structure of the Zipf's Law approach, yet diverges in its focus on calculating the number of words per text.

The mentioned values led us to establish a threshold of 450.303. With this threshold in place, the prediction process commenced: texts with word counts surpassing it were predicted as 0, indicating

	label 0	label 1
min	2	6
max	33220	2665
mean	583.755	316.850

Table 2: Word counts of labels 0 and 1

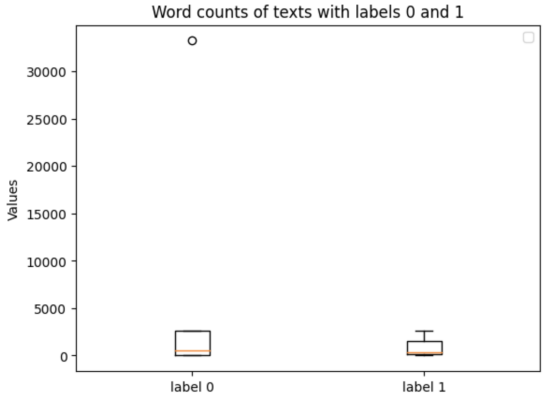


Figure 2: Word counts of labels 0 and 1

a human author, while those falling below were labeled as 1, indicating machine-generated text. Subsequently, we proceeded with text extraction for the test data, calculating the word counts for each text. We made predictions based on this.
Our preliminary F1 score for this approach stood at 0.59 on the development set, which, while respectable, fell short of the performance achieved by the Zipf's Law approach. This discrepancy led us to opt against pursuing further development of the unigram approach.

3.3 Comparing Zipf's Law and Unigram

The unigram method focuses on capturing information related to word frequencies, providing insights into the overall lexical diversity and richness of the text. In contrast, the Zipf method leverages the distributional characteristics of word frequencies, emphasizing patterns of occurrence and rank-order relationships. Together, these approaches offer complementary perspectives on textual content, enabling a more comprehensive analysis of linguistic features.
The unigram method may excel in scenarios where the distribution of word frequencies significantly impacts classification outcomes, such as detecting texts with distinct lexical signatures or vocabulary usage patterns. On the other hand, the Zipf method may prove more effective in identifying structural patterns and deviations from expected frequency distributions, particularly in texts generated by language models with predictable language patterns. While the unigram and Zipf methods differ in their primary focus and underlying principles, there is some overlap in the information they capture. The unigram method operates at the level of individual word occurrences, providing insights into the frequency and distribution of specific terms within the text, while the Zipf method considers the broader distributional patterns of word frequencies, focusing on rank-order relationships and overall distribution shapes.

4 Experimental Setup

For implementing the Zipf's Law and word unigram approach, we relied on Counter Python Software Foundation (2022), numpy NumPy (2022), and curve_fit from scipy.optimize SciPy (2022), without requiring any additional external tools or libraries. We utilized the provided data without creating additional splits. During testing on the development data, we employed the function f1_score from sklearn.metrics scikit-learn (2022).

5 Results

Our initial two approaches exhibit commendable performance in accurately predicting labels. The F1 score for the labels predicted by the Zipf's Law approach was 0.729 in the official ranking, with the task organizers' baseline set at 0.884. Our submission secured the 83rd position out of 137 in the official rankings. The preliminary F1 score for the unigram approach was 0.60, reflecting the test phase; however, this result was not included in the final submission.

6 Conclusion

In conclusion, we are satisfied with the performance of our Zipf's Law system in the shared task, particularly given its simplicity compared to other model-based approaches. The unigram system demonstrated commendable performance as well. We also explored training a linear Support Vector Classifier (SVC) scikit-learn (n.d.a) using character n-grams and employing a sublinear tf-idf scikit-learn (n.d.b) approach. We integrated several models, partitioning the training data into distinct files representing specific models used, such as ChatGPT, Cohere, Davinci, and Dolly, ensuring a balanced distribution of human-generated texts across all model categories. Despite our careful

preparations, all four models unexpectedly produced identical labels for all texts during prediction, rendering the system ineffective. This unexpected outcome highlights the necessity of rigorous testing and debugging to ensure the reliability of our methods. Identifying and resolving the underlying issues will be crucial for future improvements in model performance and credibility. Moving forward, we intend to enhance both the Zipf's Law and unigram systems through a comprehensive review of relevant literature. Additionally, we're dedicated to fixing the bug in our tf-idf vectorizer to maximize its potential in future iterations.

References

BigScience. n.d. Bloomz. Hugging Face. Accessed: February 13, 2024.

Cohere. n.d. Cohere. Cohere. Accessed: February 13, 2024.

Hugging Face. n.d. Databricks/dolly-v2-12b. Hugging Face. Accessed: February 13, 2024.

Guido M. Linders and Max. M. Louwerse. 2020. Zipf's law in human-machine dialog. In *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents*, IVA '20, New York, NY, USA. Association for Computing Machinery.

Hoang-Quoc Nguyen-Son, Ngoc-Dung T. Tieu, Huy H. Nguyen, Junichi Yamagishi, and Isao Echi Zen. 2017. Identifying computer-generated text using statistical analysis. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1504–1511.

NumPy. 2022. NumPy: Array processing for numbers, strings, records, and objects. Accessed: February 13, 2024.

OpenAI. 2022. ChatGPT. AI language model. Accessed: February 13, 2024.

OpenAI. n.d. DaVinci. DaVinci. Accessed: February 13, 2024.

Python Software Foundation. 2022. Python Collections Module. Accessed: February 13, 2024.

scikit-learn. 2022. scikit-learn: Machine Learning in Python. Version 1.0.2.

scikit-learn. n.d.a. Linear SVC. scikit. Accessed: February 18, 2024.

scikit-learn. n.d.b. tf-idf. scikit. Accessed: February 18, 2024.

SciPy. 2022. SciPy: Scientific Library for Python. Accessed: February 13, 2024.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Acknowledgements

We extend our appreciation to Çağrı Çöltekin for his assistance during this shared task.

Team Unibuc - NLP at SemEval-2024 Task 8: Transformer and Hybrid Deep Learning Based Models for Machine-Generated Text Detection

Teodor-George Marchitan[1,3,*], **Claudiu Creanga**[2,3,*], **Liviu P. Dinu**[1,3]
[1] Faculty of Mathematics and Computer Science,
[2] Interdisciplinary School of Doctoral Studies,
[3] HLT Research Center,
University of Bucharest, Romania
teodor.marchitan@s.unibuc.ro, claudiu.creanga@s.unibuc.ro, ldinu@fmi.unibuc.ro

Abstract

This paper describes the approach of the UniBuc - NLP team in tackling the SemEval 2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection. We explored transformer-based and hybrid deep learning architectures. For subtask B, our transformer-based model achieved a strong **second-place** out of 77 teams with an accuracy of **86.95%**, demonstrating the architecture's suitability for this task. However, our models showed overfitting in subtask A which could potentially be fixed with less fine-tunning and increasing maximum sequence length. For subtask C (token-level classification), our hybrid model overfit during training, hindering its ability to detect transitions between human and machine-generated text.

1 Introduction

Task 8 from SemEval 2024 competition (Wang et al., 2024a) aims to tackle the complex challenge of distinguishing between human and AI generated text. Doing so is crucial for maintaining the integrity and authenticity of information as it helps prevent the spread of misinformation and ensures that content sources are accountable. By developing tools for this task, which work in a multilingual setting, and releasing them open source we can combat non-ethical uses of AI such as propaganda, misinformation, deepfakes, social manipulation and others.

The systems developed for subtasks A and B are based on transformer models with different layers selection and merging strategies, followed by a set of fully connected layers. The training is split in two phases: a) freezing phase, where the transformer weights are not updated, only the fully connected layers are updated with a specific learning rate; b) fine-tuning phase, where the selected layers of the transformer and the fully connected layers are updated with a different (usually smaller) learning rate. For the subtask C, a different architecture was used, combining character level features, extracted with a CNN model, with word embeddings and fed into a Bidirectional LSTM followed by a set of fully connected layers. The same training strategy with different learning rates was used.

Our error analysis revealed that overfitting remains a primary challenge, despite our initial precautions. We learned that for future fine-tuning of transformer models, we should dedicate a lot more time to prevent overfitting. We made our models open source in a GitHub Repository.

2 Background

The competition had 3 tasks explained below (Figure 1). Subtask A had 2 sub-tracks: monolingual (English only) and multilingual.

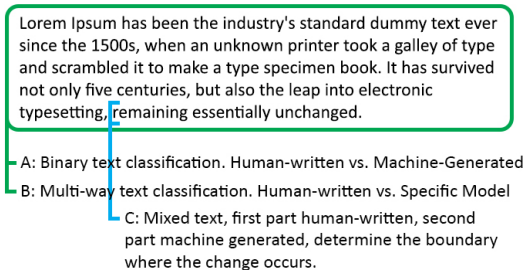


Figure 1: Three sub-tasks explained

We participated in all 3 tasks with the best result being second place on subtask B (Table 1).

2.1 Dataset

The data for this task is an extension of the M4 dataset (Wang et al., 2024b,c). Compared to subtask A and B, subtask C had much less data to work with. We found out that we could increase the size of our datasets for subtask A monolingual

[*] Equal contributors

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 403–411

	A mono	A multi	Track B	Track C
Score	85.13	79.43	**86.95**	74.28
Place	33 / 137	30 / 68	**2 / 77**	31 / 33

Table 1: Team results

	A mono	A multi	Track B	Track C
Train	119757	172417	71027	3649
Dev	5000	4000	3000	505
Test	34272	42378	18000	11123

Table 2: Datasets sizes used in this competition by tasks.

by adding the dataset from subtask B and remove duplicated items (Table 2).

2.2 Previous Work

Since GPT-2, it has been particularly difficult to detect machine-generated text, such that classical machine learning methods can no longer help. Previously, when models used top-k sampling, this resulted in text filled with too many common words and models could detect this anomaly easily (Ippolito et al., 2020). But now with bigger and bigger models and other type of sampling (like nucleus sampling), fewer artifacts are left for a detector to spot. Solaiman et al. (2019) showed that by fine-tuning a RoBERTa model we can achieve state of the art results for GPT-2 generated text with a 95% accuracy.

If for GPT-2, expert human evaluators achieved an accuracy of 70% (Ippolito et al., 2020), for GPT-3 and later models their accuracy is on par with random chance (Clark et al., 2021). It is still an open question if we can improve automated detection. Many companies (like OpenAI and Turnitin) are releasing products and claim to do it, but suffer from low rates of accuracy. In July 2023, OpenAI removed its product for this reason.

3 System overview

In this paper, we focused our research on two different system architectures: **Transformer based models** (3.1) and **Hybrid deep learning models** (3.2).

Both architectures use a block of fully connected layers (Figure 2) with the base structure being initiated with a linear layer, succeeded by normalization, a tanh activation function, followed by a dropout layer (0.5). Finally, it concludes with a linear layer with an output size of 1 for subtask A and 6 for subtask B .

Figure 2: Fully connected layer base structure

3.1 Transformer based models

The core of this architecture is based on transformer models (Figure 3). The strategy is to use the transformer model as a feature extractor, pass the information through fully connected layers (Figure 2) and apply the activation function based on the predictions for each task.

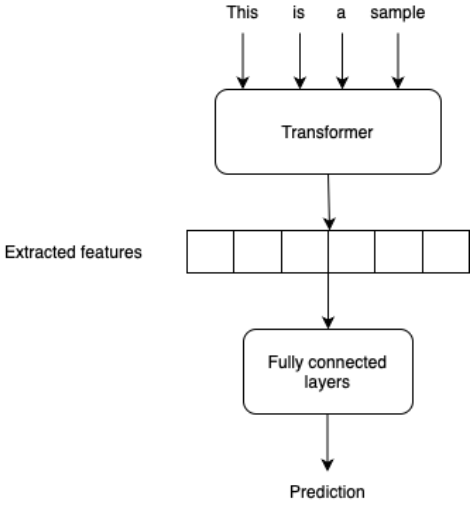


Figure 3: Transformer based models architecture

During the process of developing our system with this architecture, we encountered three difficulties that we had to address: 1) Long texts but limited number of tokens accepted by the transformer models (3.1.1); 2) Layer selection for feature extraction step (3.1.2); 3) Fine-tuning strategy to prevent overfitting (3.1.3).

3.1.1 Long text problem

Most of the transformer models accept a maximum of 512 tokens per sequence. We have also experimented the same strategies as described by Sun et al. (2020) in order to handle long texts.

I. **Truncation methods:**

- **Head only**: Keep only the first 510 tokens from the entire text. (extra 2 tokens for [CLS] and [SEP] tokens)

- **Tail only**: Keep only the last 510 tokens from the entire text. (extra 2 tokens for [CLS] and [SEP] tokens)
- **Head and Tail**: Combined the first 128 tokens with the last 384 tokens from the entire text.

II. **Hierarchical methods:** Each text is split into $k = L/512$ chunks. For each chunk we get the pooled representation of [CLS] token and merge all chunk representations using mean or max.

Our experiments proved that truncation method with **head only** works best for the given dataset as well.

3.1.2 Layer selection

Most transformer models have multiple layers and each layer is capturing different features from the input text (Sun et al., 2020). Intuitively, lower layers capture more general features at the token level and as we move up the layers, the captured features are more contextualized and more sensitive to the context of the tokens.

From our experiments, concatenating the last 4 layers and using only the last layer from the transformer proved to give the best results. Because of the limited resources, we chose to use only the last layer.

3.1.3 Fine-tuning strategy

Fine-tuning the transformer model for a downstream task is also challenging. Each layer of the transformer captures a different level of semantic and syntactic information from the input text (Yosinski et al., 2014; Howard and Ruder, 2018; Sun et al., 2020). We implemented a Head-First Fine-Tuning (HeFit) strategy (Michail et al., 2023) and used different learning rates for different layers (Sun et al., 2020):

1. For the first number of epochs $[1,\ k]$ we completely freeze the transformer layers without updating any of the weights.

2. For the rest of the epochs $[k+1,\ N]$ we fine-tune only the selected layers used for feature extraction.

Using this strategy, we are not only using less resources, but we can also preserve the more general information of the transformer (freezing lower layers) and updating information that is most relevant to the downstream task (fine-tuning selected upper layers).

3.2 Hybrid deep learning models

This model architecture (Figure 5) was inspired by the work of Chiu and Nichols (2016) which proved to be very efficient for named entity recognition task. The idea was to convert words and characters into vector representations using lookup tables and concatenate them in order to be fed into a neural network. For the character-level features we used a lookup table for the character embeddings and applied a 1D convolution followed by a 1D max pooling layer (Figure 4). For the word-level features we used a lookup table for the word embeddings. We concatenated the word and character features and fed them through a bidirectional LSTM and then a set of fully connected layers (Figure 5 - method 1).

This model was mainly used for the subtask C, which we treated as a token classification task. Therefore we have also made some experiments adding a conditional random field (Sutton and McCallum, 2010) on top of the fully connected layers (Figure 5 - method 2). This method was proved to be very efficient for sequence tagging by the work of Huang et al. (2015).

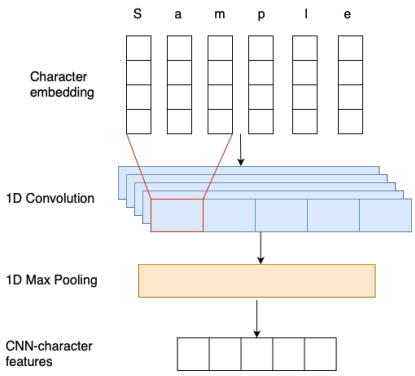


Figure 4: CNN-character level features

3.3 Experimental setup

During the training phase, we utilized the development (dev) dataset as our test set, while the training dataset was divided into a training subset and a validation subset, following an 80%-20% split. For the construction of the final model, the entire training dataset was used for training purposes, with the dev set serving as our validation set. In terms of text preprocessing, we experimented with

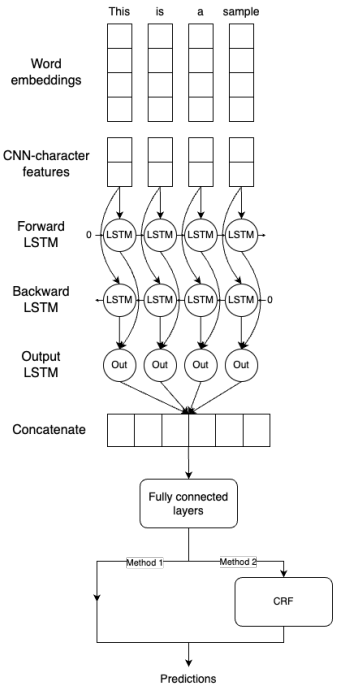


Figure 5: Hybrid deep learning model architectures. Method 1 to use the predictions directly from the fully connected block and method 2 using CRF before predictions.

three different approaches:

- **Heavy**: Involved removing pre-trained language model special tokens such as <pad>, <s>, <unk>, etc., converting numbers into words, and eliminating special characters or formats like emails and URLs.
- **Light**: Consisted of converting text to lowercase and removing special characters, including numbers.
- **None**: Text was used as is, without any preprocessing.

We observed that the model performed best with no preprocessing, a finding that aligns with the inherent flexibility of Masked Language Models to efficiently process raw text.

To determine the optimal number of training epochs, both when the pre-trained layers were kept frozen and during the fine-tuning phase, we monitored the validation set's loss and the test set's performance, opting for conservative epoch counts to prevent overfitting.

3.4 Subtask A

For this subtask, in order to be able to run the models based on the transformer architectures, we used the head only truncation strategy (3.1.2 - I.) with the first 512 tokens.

3.4.1 Monolingual

In the monolingual track, the final submission is a transformer-based model architecture (3.1) with RoBERTa-base pre-trained model. The extracted features from the transformer are only from the $[CLS]$ token of the last hidden layer with a 0.3 dropout applied. The fully connected block is built with 2 base structures (Figure 2) consisting of $[256, 64]$ neurons. A 0.5 dropout is applied and $sigmoid$ activation function is used in order to make the predictions. We trained this model in total for 5 epochs with the entire transformer architecture freezed and a batch size of 24 using the AdamW optimizer with a learning rate of $2e-4$ and the binary-cross entropy loss.

Regarding the layer selection, most of the experiments were done only using the last layer. We did some testing with last 4 layers (for some pre-trained transformers) but we could not batch size 24 anymore because of the limited resources if it were to also fine-tune the transformer's selected layers. We have also tested with multiple batch sizes and 24 seemed to work best in our case. Results in Table 4.

3.4.2 Multilingual

For the multilingual track we used models pre-trained in a multilingual context (Table 3) and for the final submission we chose mdeberta-v3-base which, even though it didn't support Indonesian, it gave the best results. The hyper-parameters that we chose were: batch size of 32, token max length of 512, a fully connecter layer (Figure 2) of 128, learning rate for the "frozen step" of 0.001 (where we train only the output layer) and smaller for fine-tuning: 0.0002.

3.5 Subtask B

In the subtask B, the final submission is a transformer-based model architecture (3.1) with RoBERTa-base pre-trained model. The extracted features from the transformer are only from the $[CLS]$ token of the last hidden layer with a 0.3 dropout applied. The fully connected block is built with 2 base structures (Figure 2) consisting of $[512, 128]$ neurons and the final output size of the model being 6. A 0.5 dropout is applied with no activation function for making the predictions. We trained this model in total for 8 epochs, first

6 epochs with the entire transformer architecture freezed, and the last 2 epochs also fine-tuning the last layer of the transformer (3.1.3). The batch size used was 32 and optimizer AdamW with a learning rate of $3e-4$ for the freeze part of the training (updating only the fully-connected block weights) and $2e-4$ for the fine-tuning part with a linear scheduler with 50 warmup steps and cross entropy loss.

Regarding the layer selection, most of the experiments were done only using the last layer. We did some testing with last 4 layers (for some pretrained transformers) but we could not batch size 32 anymore because of the limited resources if it were to also fine-tune the transformer's selected layers. We have also tested with multiple batch sizes and 32 seemed to work best in our case. Results in Table 5.

3.6 Subtask C

We treated this subtask as a token classification one and changed the labels from positions to list of 0 and 1, where 0 means that the token at that specific position is not machine generated and 1 otherwise.

The tokenization was done by splitting the text by space and kept only the first 1024 tokens from the entire text. As for the maximum character length of the tokens we went with 25.

The final submission is a hybrid deep learning model architecture (3.2). We used the method 2 variation of the architecture (Figure 5 with the CRF model right before making the predictions.

For the CNN-character features we set the character embeddings dimension to 10 and randomly initialized the lookup table using uniform distribution with range $[-0.5, 0.5]$. We used the convolution with kernel size 3 and 20 filters with a 0.5 dropout afterwards. The word embedding dimension used is 300 and the lookup table randomly initialized in the same manner. For the bidirectional LSTM we used 2 filters with 32 hidden dimension each. The fully connected block is build with a fully connected base structure (2) with 32 neurons and final output size of 2.

We trained this model in total for 3 epochs with a batch size of 12 and optimizer AdamW with a learning rate of $5e-3$ for the first 2 epochs of the training and $3e-3$ in the last epoch together with a linear scheduler with no warmup steps and cross entropy loss.

4 Results

4.1 Subtask A

For both monolingual and multilingual our model under-predicted the human-written class. In the case of the monolingual track our model performs equally well in detecting machine-generated text for each model, but under calls the negative class (Figure 6). It predicts 23043 items as machine generated and 11229 as human-written while the truth was more balanced (18000 vs. 16272). We obtain good accuracy for each machine generated model, but we under-call the human label (0.68 accuracy) so in the end the final score is 0.85.

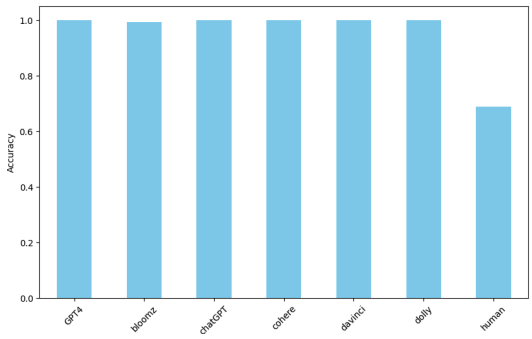


Figure 6: Subtask A: monolingual - accuracy by model for test set

In the case of multilingual, testing on dev data gave us an accuracy of 0.96 but the final test score was 0.79. Our model predicted 30764 samples as machine generated and only 11614 as human-written, while the true distribution was more balanced (22140 vs. 20238). This suggests that our model was overfit and had a bias for the positive class. If we look at the distribution per model we can see that we have a good accuracy on all models, except for human and a bit worse for chatGPT (Figure 7), ending up with a final score of 0.79.

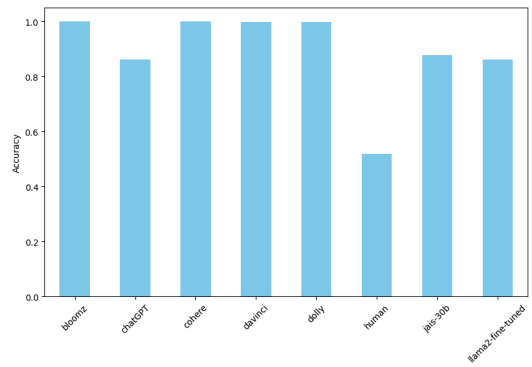


Figure 7: Subtask A: multilingual - accuracy by model for test set

If we look at sequence length we can see an U shaped graph at 500 - 1500 number of tokens, where the model performs worst (Figure 8) for both monolingual and multilingual tracks. We believe this is because our transformers had a limit of 512 for token length and we didn't have the resources to train on a bigger sequence length.

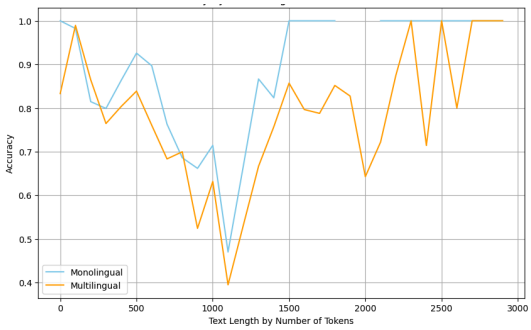


Figure 8: Subtask A: accuracy by sequence length in tokens, monolingual and multilingual

4.2 Subtask B

Our most notable performance was achieved in subtask B, where we secured the **second position** from a total of 77 participating teams, with an accuracy score of **86.95%**, very close to first position. Upon examining the accuracy breakdown by model, it becomes evident that our model exhibited strong performance, particularly with bloomz and chatGPT outputs, while facing more challenges with cohere (refer to Figure 9). The elevated score compared to Task A implies that our model's architecture and training methodology were well-suited for the demands of a multiclass classification task.

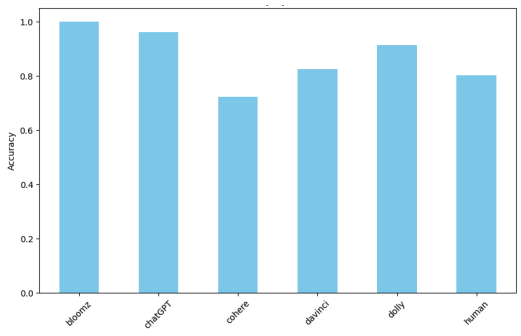


Figure 9: Subtask B: accuracy by model for test set.

4.3 Subtask C

Our results on the subtask C show that the model architecture we chose alongside the hyperparameters overfitted drastically on this dataset. The MAE on training data decreased from 18.8 to 4.39 and on validation data decreased from 18.04 to 8.34 during the training phase, while on the final test dataset the MAE increased to 74.28. This proves that the character and word embeddings could not generalize that good in order to be able to find that transition spot from human text to machine generated text.

5 Conclusions and Future Work

In conclusion, our architecture and training methods produced good results for subtask B (securing the second place). However, our models demonstrated signs of overfitting for subtask A. We could not find a proper explanation for why the model architecture work better on subtask B and is overfitting that much on the other task. Our future endeavors will explore several avenues:

- **Extended Sequence Lengths**: With more powerful machines we plan to increase the token length from 512 to 1024 in order to capture a wider context, which could improve their performance.
- **Ensemble Learning with Model Specialization**: Split the dataset by originating model (chatGPT, cohere etc.) and train specialized models on each subset. Each specialized model will become adept at discerning text generated by its corresponding model. By aggregating predictions from these specialized models, we aim to construct a meta-model capable of making better final predictions.
- **LLM**: We plan to investigate the efficacy of large language models (like Mistral/Mixtral or Solar) with either zero shot learning or few shot learning scenarios. For few-shot learning, we intend to exploit the in-context learning capabilities of LLMs by presenting them with pairs of examples (one human-written and one machine-generated) within the same context window. We will then ask the model to predict an unseen example.

Acknowledgements

This work was partially supported by a grant on Machine Reading Comprehension from Accenture Labs and by the POCIDIF project in Action 1.2. "Romanian Hub for Artificial Intelligence".

References

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Andrianos Michail, Stefanos Konstantinou, and Simon Clematide. 2023. Uzh$_clypatsemeval - 2023task9 : Head - firstfine - tuningandchatgptdatageneration for cross - linguallearningintweetintimacyprediction$. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *ArXiv*, abs/1908.09203.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification?

Charles Sutton and Andrew McCallum. 2010. An introduction to conditional random fields.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Akim Tsvigun, Jinyan Su, Artem Shelmanov, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024c. MG-Bench: Evaluation benchmark for black-box machine-generated text detection.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?

A Further experiments - Subtask A

For most of the experiments in subtask A monolingual, we used two fully connected layers (2) with $[256, 64]$ neurons, batch size 24 and trained the model in total for 5 epochs. For all experiments we used AdamW optimizer with learning rate $2e{-}4$ and binary-cross entropy loss. For some of the experiments we have also tried fine-tuning the last n selected layers (in most cases just the last layer) for the last k epochs. In those cases, we have also used a linear scheduler with 50 warmup steps and changed the learning rate as well. The results can be seen in Table 4. Experiments for the multilingual track kept the same architecture as the monolingual one but used multilingual pre-trained models Table 3.

Model	Train	Validation	Test	Final
mdeberta-v3	0.96	0.95	0.94	**0.79**
xlm-roberta	0.97	0.95	0.92	0.78
bert-multi	0.95	0.92	0.91	0.75
distilbert-multi	0.93	0.90	0.89	0.73

Table 3: Experiment results by pre-trained model - multilingual. Validation was the dev set, test size was 0.2 and final score is the test score in competition.

B Further experiments - Subtask B

For most of the experiments in subtask B, we used two fully connected layers (2) with $[512, 128]$ neurons, batch size 32 and a trained the model in total 8 epochs. For all experiments we used AdamW optimizer with learning rate $3e - 4$ and cross entropy loss. For some of the experiments we have also tried fine-tuning the last n selected layers (in most cases just the last layer) for the last k epochs.

In those cases, we have also used a linear scheduler with 50 warmup steps and changed the learning rate as well. The results can be seen in Table 5.

Base model	Epochs before fine-tune	LR fine-tune	Train	Validation	Test	Final
roberta-base	5	—	0.89	0.94	0.89	**0.85**
flan-t5-base	5	—	0.98	0.97	0.95	0.84
deberta-v3-large	5	—	0.98	0.97	0.96	**0.85**
albert-base-v2	5	—	0.77	0.82	0.74	0.83
bert-base-cased	5	—	0.79	0.80	0.76	**0.86**
distilbert-base-uncased	5	—	0.84	0.85	0.79	0.74
gpt2	5	—	0.92	0.92	0.86	0.76
xlm-roberta-base	5	—	0.74	0.79	0.75	0.83
xlnet-base-cased	5	—	0.74	0.80	0.79	0.79
roberta-base	4	0.0002	0.88	0.92	0.88	0.83
roberta-base	3	0.0001	0.99	0.99	0.93	0.68

Table 4: Experiment results for Subtask A - monolingual track. Validation was the dev set, test size was 0.2 and final score is the test score in competition.

Base model	Epochs	Epochs before fine-tune	LR fine-tune	Train	Validation	Test	Final
roberta-base	8	6	0.0002	0.98	0.97	0.90	**0.87**
roberta-base	6	6	—	0.76	0.86	0.74	0.59
bert-base-cased	8	6	0.0002	0.92	0.88	0.90	0.57
bert-base-cased	6	6	—	0.67	0.76	0.63	0.47

Table 5: Experiment results for Subtask B. Validation was the dev set, test size was 0.2 and final score is the test score in competition.

LinguisTech at SemEval-2024 Task 10: Emotion Discovery and Reasoning its Flip in Conversation

Mihaela Alexandru
Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Romania
alexandrumihaela227@gmail.com

Călina-Georgiana Ciocoiu
Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Romania
calinaciocoiu@gmail.com

Ioana Măniga
Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Romania
ioana.mna@gmail.com

Octavian Ungureanu
Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Romania
tavi2105@gmail.com

Daniela Gîfu
Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Romania
Institute of Computer Science, Romanian Academy - Iasi Branch
daniela.gifu@iit.academiaromana-is.ro

Diana Trandăbăț
Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi, Romania
dtrandabat@info.uaic.ro

Abstract

The "Emotion Discovery and Reasoning Its Flip in Conversation" task at the SemEval 2024 competition focuses on the automatic recognition of emotion flips, triggered within multi-party textual conversations. This paper proposes a novel approach that draws a parallel between a mixed strategy and a comparative strategy, contrasting a Rule-Based Function with Named Entity Recognition (NER)—an approach that shows promise in understanding speaker-specific emotional dynamics. Furthermore, this method surpasses the performance of both DistilBERT and RoBERTa models, demonstrating competitive effectiveness in detecting emotion flips triggered in multi-party textual conversations, achieving a 70% F1-score. This system was ranked 6th in the SemEval 2024 competition for Subtask 3.

1 Introduction

The field of emotion analysis continues to be rich with surprises (Kumar et al., 2022), especially within the context of conversations. For this competition, we have implemented a competitive method for Subtask 3 (Kumar et al., 2024). Uncovering the reasons (triggers) behind a speaker's emotional shift during a conversation—taking the example of "Friends," an American television sitcom—presents a unique challenge, especially in the realm of response generation (Gifu and Cioca, 2013). With the rising popularity of chatbots (Ouatu et al., 2020), it appears that emotions are the critical link missing between establishing trust and simulating genuine connections (Madasu et al., 2023). Furthermore, the detection of emotions and their triggers (Cristea et al., 2015) could play a significant role

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 412–419

in new digital marketing strategies, enhancing user feedback, and analyzing overall customer centricity.

This raises a pertinent question: *Is AI capable enough to identify emotions and their triggers with high accuracy within code-mixed dialogues?*

The remainder of this paper is organized as follows: Section 2 briefly reviews studies related to emotion recognition (Kumar et. al., 2023) and the concept of an emotion flip in conversations. Section 3 describes the system developed to detect the specific emotional dynamics that occur during a conversation. Section 4 outlines the experimental setups. Section 5 discusses the results of the experiments conducted, and Section 6 presents the conclusions.

2 Background

Recent research in dialogue emotion detection has witnessed significant advancements. The literature suggests that the challenge of recognizing emotions in conversations can be tackled from various perspectives. For instance, a notable approach involves the use of models based on transformers, as well as iterative emotion interaction networks.

The most prevalent method for emotional discovery and analysis in recent years involves employing various transformers. Variants of BERT have been frequently utilized, whether they are pre-trained or not. Some of the notable examples include mBERT (De Bruyne et al., 2022), LFTW-RoBERTa, YT-Bert, MNLI-BART-large, MNLI-RoBERTa (Bulla et al., 2023), and EmoRoBERTa (Bayram & Benhiba, 2022), among others. Additionally, a study by Li et al. (2020) introduced HiTrans, an innovative model specifically designed to discern emotions within multi-speaker conversations. A team of researchers (Kumar et al., 2023) has presented a pioneering approach that focuses on identifying the triggers behind emotion shifts in conversations. Using BERT as a foundation, their findings indicate that TGIF (a novel neural architecture) more effectively addresses the increase in instigator labels compared to existing baselines. Some studies concentrate on the application of zero-shot models to emotion classification and hate speech detection (Bulla et. al., 2023), while others adopt a modified approach, developing a semi-zero-shot model. This variation aims to investigate and determine whether significant challenges and differences exist in emotion detection across various language families (De Bruyne et al., 2022). Interestingly, the F1-scores for all transformer types employed in zero-shot scenarios are reported to be similar across both studies.

In the experiments dedicated to the KET model (Zhong et al., 2019), several key findings were highlighted: notably, the KET model demonstrated superior performance, surpassing existing state-of-the-art models in various datasets as measured by F1 score. This underscores its effectiveness in detecting emotions within textual conversations. Additionally, there is research (Lu et al., 2020) exploring non-transformer-based solutions, such as the innovative Iterative Emotion Interaction Network. This approach specifically addresses the challenge of the absence of gold-standard emotion labels during inference, offering a novel solution to a prevalent issue in emotion detection.

Additional research (Zhu et al., 2021) explores the use of baselines such as DialogueGCN and KET, but it is COSMIC that emerges as the superior model among these baselines. This advancement began with the development of a topic-augmented language model (LM), which includes a dedicated layer for detecting topics. These collective efforts significantly push the boundaries of dialogue emotion detection forward by incorporating a blend of knowledge, contextual insight, and cutting-edge neural architectures.

The third subtask of SemEval-2024 Task 10, titled 'Emotion Discovery and Reasoning its Flip in Conversation' (EDiReF), is dedicated to exploring the point in a dialogue at which the last emotion flip occurs For the Emotion Flip Reasoning subtask, Task 10 of SemEval-2024 provides three types of datasets: training, validation, and testing, detailed in the table below:

Training Dataset	Validation Dataset	Testing Dataset
400 entries	426 entries	1002 entries
13500 dialogue lines	3522 dialogue lines	8642 dialogue lines

Table 1: Task Dataset Statistics

The datasets contain dialogues extracted from different episodes of the 'Friends' series, stored in a JSON array. Each entry comprises the following fields:

- episode: the name of the episode (e.g. "episode": "utterance_0");
- speakers: a list of speakers in order of their participation in the conversation (e.g. "Chandler", "The Interviewer", "Chandler", "The Interviewer", "Chandler");
- emotions: a list of emotions in order (e.g. "neutral", "neutral", "neutral", "neutral", "surprise",);
- utterances: the list of utterances from the dialogue in sequential order (e.g. "also I was the point person on my company's transition from the KL-5 to GR-6 system.", "You must've had your hands full.", "That I did. That I did.", "So let's talk a little bit about your duties.", "My duties? All right.");
- triggers: a list of triggers in sequential order. This field is the output of our models and represents a list of '0.0s' and only one value of '1.0', indicating the trigger in that conversation.

Before proceeding further, we conducted a thorough examination of the training dataset for our subtask to gain insights into the appearance of triggers and the functioning of the Emotion Flip Reasoning (EFR) system. Our analysis revealed that all triggers are associated with the same (last) emotion flip in the dialogue. Additionally, we observed that triggers can manifest in any utterance within the same segment of the conversation where the emotion change occurs. To achieve this understanding, we initially examined the speakers, emotions, and triggers. Subsequently, we delved into the utterances, particularly focusing on cases where triggers were less clear. As observed in numerous papers, the implementation of models often revolves around transformers, with BERT being a prominent choice. This observation significantly influenced our approach, leading us to adopt a strategy centered on utilizing the DistilBERT transformer. DistilBERT, developed to reduce the size and enhance the computational efficiency of BERT while preserving a substantial portion of its functionality (Sanh et al., 2019), emerged as a key component of our investigation. Additionally, we incorporated the RoBERTa transformer into our architecture's model, reflecting our commitment to leveraging state-of-the-art techniques. This initiative can be seen in the baseline part of our architecture model.

3 System Overview

Our objective is to enhance emotion recognition technology by investigating the underlying reasons for sudden emotional changes. Specifically, our research concentrates on emotional flips, which denote abrupt shifts in emotions during conversation—an aspect often overlooked in existing studies. Despite the progress achieved by previous methods, recognizing emotions in conversation remains challenging due to the nuanced conveyance of emotions and the varying significance of utterances, influenced by the specific topics discussed and implicit understandings shared among participants.

Upon analyzing the dataset, we identified seven distinct emotion labels: neutral, joy, surprise, anger, sadness, fear, and disgust, with varying frequencies. Dialogues in the dataset involve a range of one to eight participants, with dialogues between two speakers being the most common.

The primary focus of this paper is to identify speaker-specific emotional dynamics occurring during conversation. Our approach utilizes two transformer-based baselines, RoBERTa and DistilBERT. Additionally, we compare their performance with a mixed and comparative method employing rule-based and Named Entity Recognition (NER) techniques.

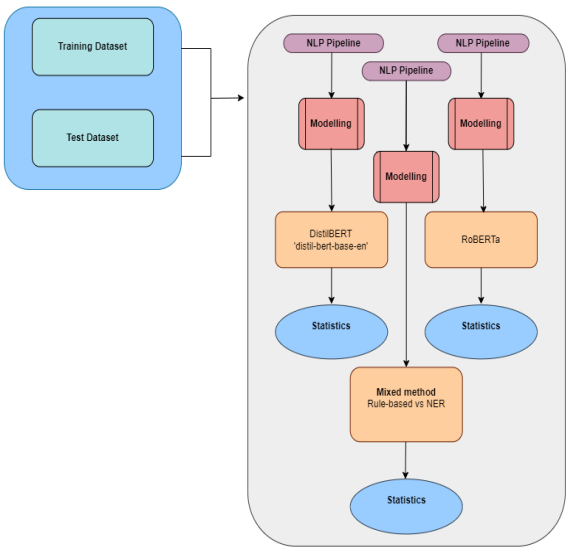


Figure 1: The LinguisTech system architecture

The first transformer baseline we utilized was a pre-trained RoBERTa model (TFRobertaModel) based on the BERT-base architecture. This model is described by: 2-layer, 768-hidden, 12-heads, 125M parameters. As for the parameters, we configured the model with the following settings:

- metrics=['acc', f1_m, precision_m, recall_m]
- loss='sparse_categorical_crossentropy'
- optimizer=tf.keras.optimizers.Adam(lr=1e-5)

In addition, we employed 'relu' and 'softmax' as activation functions. We segmented each conversation into utterances, and for each utterance, the training data is structured as a dictionary containing the following fields:

- utterance – the current utterance
- emotion - the current emotion
- context – containing arrays with: all emotions in that dialog, all speakers, all utterances

In the pre-processing phase for the RoBERTa baseline, we pursued several approaches and actions:

- Extracted all replicas from the context and applied tokenization, lemmatization, stopword removal, etc.
- Extracted emotions from contexts.
- Extracted emotions and utterances from context.
- Extracted emotions, utterances, and speakers from context.
- Retained the context along with the following: id, list of utterances, list of emotions, list of speakers.
- Retained the context along with individual replicas, list of utterances, list of emotions, list of speakers.
- Maintained the original context while eliminating the first half, followed by attempting to remove the first half of the context and combining speakers, emotions, speakers, and emotions.

As for the second baseline model, we chose the DistilBertClassifier from the keras_nlp framework. We utilized the 'distil_bert_base_en' preset, which is a 6-layer DistilBERT model maintaining case sensitivity. This model comprises 65.19 million parameters and was trained on English Wikipedia + BooksCorpus using BERT as the teacher model. For parameters, we configured the model with the following settings:

- loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True)
- optimizer=keras.optimizers.Adam(5e-5)
- jit_compile=True
- metrics=['accuracy', f1_m, precision_m, recall_m], where f1, precision and recall are functions defined by us with the traditional method.

In the preprocessing phase for the DistilBERT baseline, we divided each conversation into utterances. For each utterance, the training data is structured as a dictionary containing the following fields:

- entry_index - the index of the utterance in conversation
- entry – a string representing the intervention of index entry_index, formed from entry_index - speaker - utterance - emotion
- context – a string formed by concatenating the entire conversation, every dialogue line being formed with this rule: speaker: utterance – emotion

After preprocessing, we applied a DictVectorizer from sklearn to convert the data into a numerical format. Additionally, we performed feature selection by selecting the 100 best features using SelectKBest (also from sklearn), with the chi-square test as the scoring function.

Examples of preprocessed data objects for RoBERTa and DistilBERT can be observed in the first and second annexes, respectively.

4 Experimental Setup

Based on the results obtained from implementing the two transformers, RoBERTa and DistilBERT, we observed outcomes that did not meet our expectations. Consequently, we initiated an experimental investigation aimed at combining and comparing two alternative methods to achievestyle

improved performance. These methods include a rule-based function constructed from observations on the dataset, as well as a Named-Entity Recognition (NER) Model.

Our initial observation revealed that triggers are generally present in the second part of the conversation. To validate our hypothesis, we calculated the instances where this statement holds true, as well as the percentage of cases where it does not. The results are as follows:

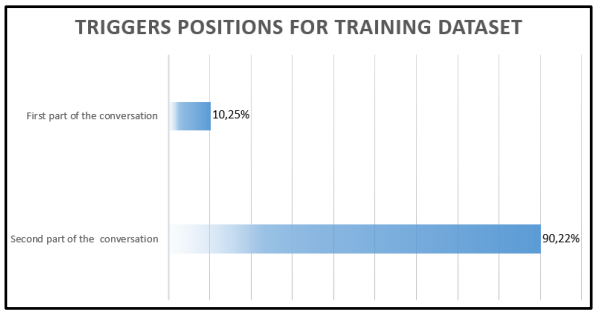


Figure 2: Trigger positions for training dataset in first/second part of conversation

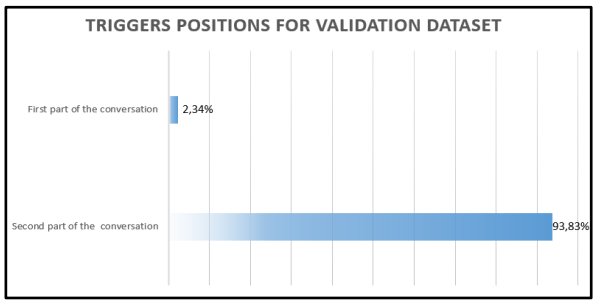


Figure 3: Trigger positions for validation dataset in first/second part of conversation

Having said that, the first rule we applied focused solely on the second part of each conversation.

The second rule is based on the emotion flips observed for each speaker. Whenever a change in emotion occurs between two consecutive interventions by a speaker, we designate the utterance preceding the second intervention as a trigger.

	Speaker	Utterance	Emotion	Trigger
1	Chandler	Hey, Mon.	Neutral	0
2	Monica	Hey-hey-hey. You wanna hear something that sucks.	Neutral	0
3	Chandler	Do I ever.	Joy	0
4	Monica	Chris says they're closing down the bar.	Sadness	0
5	Chandler	No way!	Surprise	1
6	Monica	Yeah, apparently, they're turning it into some kind of coffee place.	Neutral	0

Table 2: Dialogue example for the second rule detected

For the NER method, we utilized TFAutoModelForTokenClassification from python library transformers library with the 'bert-base-cased' preset.

As for the parameters, we configured the model with the following settings:

- optimizer=tensorflow.keras.optimizers.Adam(learning_rate=2e-5)
- epochs = 3 (the best score was obtained on running with 3 epochs)
- metrics: 'precision', 'recall', 'f1', 'accuracy'
- tensorflow.keras.callbacks.EarlyStoppig(monitor='val_loss', patience=3)

From the dataset, we only used emotions and triggers from every conversation. Because the model solves a tagging problem, we arranged the attributes in two separate lists, so that there is a 1-1 correspondence between their elements. We also renamed the triggers into labels: 0.0 = 'no' and 1.0 = 'yes'. An example of preprocessed data objects for NER can be observed in the third annexe.

{

"tokens": ["neutral", "neutral", "neutral", "neutral", "surprise"],

"labels": ["no", "no", "no", "yes", "no"]

}

After that, we applied tokenization with AutoTokenizer from transformers.

We also concatenate the train and validation dataset and applied a random split on the result, with the pivot value of 80% of the dataset length, so that we use 80% for training and 20% for validation.

5 Results

Upon comparing the Rule-Based Function and Named-Entity Recognition Methods, we obtained the results (F1 score of the triggers) displayed in the following table:

	Method	Score
1	Rule-based method	0.45
2	NER model with 3 epochs - cased	0.68
3	NER model with 3 epochs with rule-based method (XOR function applied on outputs) cased	0.47
4	NER model with 1 epoch cased	0.67
5	NER model with 5 epochs cased	0.66
6	NER model with 3 epochs uncased	0.70

Table 3: Comparing Scores (Rule-Based Function – NER) methods

From the results, it is evident that the highest F1 score is achieved by submission 2, which utilized the NER model trained over 3 epochs. Interestingly, as the number of epochs exceeded 5, we observed a consistent decrease in the F1 score.

	Method	F1 Score
1	RoBERTa Baseline	0.00
2	DistilBERT Baseline	0.00
3	NER model with 3 epochs - cased	0.68
4	NER model with 3 epochs - uncased	0.70

Table 4: Comparing Scores (Baselines vs NER)

The preceding table showcases the results achieved with the various methods we applied. Notably, the method using NER with 3 epochs outperformed the others, achieving F1 scores between 0.6 and 0.7 (Training/Validation). In comparison, our implementations using baseline methods yielded lower F1 scores: sthe DistilBERT Baseline method obtained a score of 0.1811%, and the RoBERTa Baseline method achieved 0.2452% (Training/Validation). It's crucial to note that these scores were calculated using our custom-defined F1 scoring function, tailored to the traditional method. Furthermore, a 0.00% score was observed when applying a different F1 scoring approach.

6 Conclusion

In this paper, we demonstrated that employing a Named-Entity Recognition (NER) model trained over 3 epochs for emotion flip detection yields superior results compared to classical approaches such as the RoBERTa and DistilBERT baselines, as well as a rule-based strategy. Our team's mixed and comparative solution outperformed the baseline models in terms of outcomes and provided valuable insights for future research on architecture and model enhancements. Notably, our method, utilizing the NER model trained over 3 epochs, achieved the highest F1 score. However, it is crucial to note that increasing the number of epochs beyond 5 led to a consistent decrease in the F1 score. Our evaluation indicates a significant performance improvement (~60% in F1-score) compared to previous studies.

In this way, we discovered that this is a complex problem, revealing numerous intriguing avenues for further exploration. Nevertheless, it is crucial to consider the potential benefits of incorporating audio and visual support, which could lead to enhanced performance. This insight prompts us to contemplate an exciting investigation for the future.

References

Kumar, S., Shrimal, A., Akhtar, Md S., Chakraborty, T. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. In: Knowledge-Based Systems, Vol. 240, 108112, ISSN 0950-7051 https://doi.org/10.1016/j.knosys.2021.108112.

S. Kumar, S. Dudeja, M. S. Akhtar and T. Chakraborty, "Emotion Flip Reasoning in Multiparty Conversations," in *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 3, pp. 1339-1348, March 2024, doi: 10.1109/TAI.2023.3289937.

Kumar, Shivani and S, Ramaneswaran and Akhtar, Md and Chakraborty, Tanmoy 2023. From Multilingual Complexity to Emotional Clarity: Leveraging Commonsense to Unveil Emotions in Code-Mixed Dialogues. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 9638-9652 doi: 10.18653/v1/2023.emnlp-main.598

Kumar, Shivani and Akhtar, Md Shad and Cambria, Erik and Chakraborty, Tanmoy 2024. "SemEval 2024 -- Task 10: Emotion Discovery and Reasoning its Flip in

Conversation (EDiReF)". In: Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics. https://arxiv.org/abs/2402.18944

Gîfu, D. and Cioca, M. 2013. Online Civic Identity. Extraction of Features. In: Procedia - Social and Behavioral Sciences, Vol. 76, University of Piteşti Publishing House 2013, pages 366-371, Elsevier, ISSN 1844-6272, https://doi.org/10.1016/j.sbspro.2013.04.129.

Ouatu, B., Gîfu. D. 2020. Chatbot, the Future of Learning? In: Proceedings of the 5th International Conference on Smart Learning Ecosystems and Regional Development (SLERD 2020), in Ludic, Co-design and Tools Supporting Smart Learning Ecosystems and Smart Education, Springer, pages 263-268. https://api.semanticscholar.org/CorpusID:224946479.

Madasu, A., Firdaus, M., and Ekbal, A. 2023. A Unified Framework for Emotion Identification and Generation in Dialogues. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 73–78, Dubrovnik, Croatia. Association for Computational Linguistics. https://aclanthology.org/2023.eacl-srw.7/.

Cristea, D., Gîfu, D., Colhon, M., Diac, P., Bibiri, A., Mărănduc, C., and Scutelnicu, L.-A. 2015. Chapter - Quo Vadis: A Corpus of Entities and Relations. In: Language, Production, Cognition, and the Lexicon. Text, Speech and Language Technology, Part VI - Language Resouces and Langauge Engineering, Nuria Gala, Reinhard Rapp and Gemma Bel-Enguix (eds.), Vol. 48, New York, USA, pages 505-543. https://doi.org/10.1007/978-3-319-08043-7_28

De Bruyne, L, Singh, P., De Clercq, O., Lefever, E., Hoste, V. 2022. How Language-Dependent is Emotion Detection? Evidence from Multilingual BERT. In Proceedings of the 2nd Workshop on Multi-lingual Representation Learning (MRL), pages 76–85, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics. https://aclanthology.org/2022.mrl-1.7/

Bulla, L., Gangemi, A., Mongiovi', M. 2023. Towards Distribution-shift Robust Text Classification of Emotional Content. In Findings of the Association for Computational Linguistics: ACL 2023, pages 8256–8268, Toronto, Canada. Association for Computational Linguistics. https://aclanthology.org/2023.findings-acl.524/.

Bayram, U. and Benhiba, L. 2022. Emotionally-Informed Models for Detecting Moments of Change and Suicide Risk Levels in Longitudinal Social Media Data. In Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology, pages 219–225, Seattle, USA. Association for Computational Linguistics. https://aclanthology.org/2022.clpsych-1.20/.

Li, J., Ji, D., Li, F., Zhang, M., and Liu, Y. 2020. HiTrans: A Transformer-Based Context- and Speaker-Sensitive Model for Emotion Detection in Conversations. In Proceedings of the 28th International Conference on Computational Linguistics, pages 4190–4200, Barcelona, Spain (Online). International Committee on Computational Linguistics. https://aclanthology.org/2020.coling-main.370/.

Kumar, S., Dudeja, S., Akhtar, M. S., and Chakraborty, T. 2023. "Emotion Flip Reasoning in Multiparty Conversations," in IEEE Transactions on Artificial Intelligence, doi: 10.1109/TAI.2023.3289937. https://ieeexplore.ieee.org/document/10164178.

Zhong, P., Wang, D., and Miao, C. 2019. Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 165–176, Hong Kong, China. Association for Computational Linguistics. https://aclanthology.org/D19-1016/.

Lu, X., Zhao, Y., Wu, Y., Tian, Y., Chen, H., and Qin, B. 2020. An Iterative Emotion Interaction Network for Emotion Recognition in Conversations. In Proceedings of the 28th International Conference on Computational Linguistics, pages 4078–4088, Barcelona, Spain (Online). International Committee on Computational Linguistics. https://aclanthology.org/2020.coling-main.360/.

Zhu, L., Pergola, G., Gui, L., Zhou, D., and He, Y. 2021. Topic-Driven and Knowledge-Aware Transformer for Dialogue Emotion Detection. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1571–1582, Online. Association for Computational Linguistics. https://aclanthology.org/2021.acl-long.125/.

Sanh, V. et al. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR abs/1910.01108. https://arxiv.org/pdf/1910.01108.pdf

A. Appendices

Data Object After Preprocessing RoBERTa


```
{
      "utterance": "also I was the point person on
my company's transition from the KL-5 to GR-6
system",
      "emotion": "neutral"
      "speakers": [
         "Chandler",
         "The Interviewer",
         "Chandler",
         "The Interviewer",
         "Chandler"
      ],
      "utterances": [
         "also I was the point person on my
company's transition from the KL-5 to GR-6
system.",
         "You must've had your hands full.",
         "That I did. That I did.",
         "So let's talk a little bit about your duties.",
         "My duties?  All right."
      ],
      "emotions": [
         "neutral",
         "neutral",
         "neutral",
         "neutral",
         "surprise"
      ]
   }
```


B. Appendices

Data Object After Preprocessing DistilBERT


```
{
"entry_index": 0,
         "entry": "0 - Chandler - also I was the point
person on my company's transition from the KL-5 to
GR-6 system. - neutral",
        "context":
"Chandler: also I was the point person on my company's
transition from the KL-5 to GR-6 system. – neutral
The Interviewer: You must've had your hands full. –
neutral
Chandler: That I did. That I did. – neutral
The Interviewer: So let's talk a little bit about your
duties. – neutral
Chandler: My duties?  All right. - surprise"
         }
```


C.Appendices

Data Object After Preprocessing NER


```
{
"tokens": [ "neutral", "neutral", "neutral", "neutral",
"surprise"],
"labels": [ "no", "no", "no", "yes", "no"]
}
```

Text Mining at SemEval-2024 Task 1: Evaluating Semantic Textual Relatedness in Low-resource Languages using Various Embedding Methods and Machine Learning Regression Models

Ron Keinan
Department of Computer Science, Jerusalem College of Technology, Lev Academic Center
21 Havaad Haleumi St., P.O.B. 16031, 9116001 Jerusalem, Israel
ronke21@gmail.com

Abstract

In this paper, I describe my submission to the SemEval-2024 contest. I tackled subtask 1 - "Semantic Textual Relatedness for African and Asian Languages". To find the semantic relatedness of sentence pairs, I tackled this task by creating models for nine different languages. I then vectorized the text data using a variety of embedding techniques including doc2vec, tf-idf, Sentence-Transformers, Bert, Roberta, and more, and used 11 traditional machine learning techniques of the regression type for analysis and evaluation.

1 Introduction

Semantic Textual Relatedness (STR), which involves determining the degree of semantic similarity or relatedness between two pieces of text, has emerged as a significant task within Natural Language Processing (NLP). This task holds significant relevance and importance across various applications, including information retrieval, question answering, and summarization. By accurately measuring the semantic relatedness between sentences, we can enhance the performance of many NLP systems and improve their overall effectiveness.

In this paper, we describe our participation in subtask 1-A of SemEval 2024, for STR of texts written in 9 languages: Algerian Arabic, Amharic, Hausa, Kinyarwanda, Moroccan Arabic, Marathi, Telugu, Spanish, and English. Our approach to solving the task was based on a previous study that dealt with a similar sentiment classification task (Keinan & HaCohen-Kerner, 2023), and was based on a comparison of different embedding methods and then a comparison between different regression classifiers. We compared the results of each classifier with other vectors and chose the model that provided the best results on the training dataset, in favor of classifying the proximity between the sentences in the test dataset.

The full description of task 1 in general and the subtasks, in particular, is given in Ousidhoum et al. (2024B), and the dataset is described in Ousidhoum et al. (2024A).

2 Background

2.1 Semantic Textual Relatedness

Semantic Textual Relatedness (STR) is pivotal in automatically assessing the semantic similarity or relatedness between pieces of natural language text, thereby offering insights into the underlying relationships between subjects(Hadj et al., 2020). STR facilitates the exploration of individuals' opinions on specific topics and enables actionable insights for future planning(Abdalla et al., 2023).

In an era marked by the proliferation of textual data across various platforms, STR serves vital purposes such as information retrieval, question-answering, and summarization. Despite the inherent complexities in STR, including nuances in language and the varying degrees of relatedness between texts, researchers are actively engaged in refining and advancing STR systems to achieve greater precision in measuring semantic textual relatedness.

Challenges abound both for computational algorithms and human evaluators in STR. Achieving accurate results in STR demands not only an understanding of linguistic context but also cultural context and specific domain knowledge(Gabrilovich & Markovitch, 2007). Budanitsky and Hirst (2006) argued that relatedness is more general than similarity, as the former subsumes many different kinds of specific relations, including opposition, functional association, and others. They claimed that computational linguistics applications often

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 420–431

require measures of relatedness rather than the more narrowly defined measures of similarity.

2.2 Semantic Textual Relatedness in Low Resources African Languages

Detecting STR in low-resource African and Asian languages poses an even greater challenge for several factors. In the realm of STR, tackling the scarcity of annotated data emerges as a significant hurdle, particularly concerning low-resource languages. Annotated data, crucial for training ML algorithms in STR, denotes text labeled with sentiments, like positive, negative, or neutral. This dearth of annotated data hampers the development of high-quality STR systems. ML algorithms thrive on ample data to discern patterns and make precise predictions. Consequently, STR systems tailored for low-resource African/Asian languages, lacking sufficient annotated data, often exhibit diminished performance and accuracy.

Moreover, the variability of sentiment expressions in low-resource African/Asian languages poses another formidable challenge. Unlike English, many languages boast a diverse palette of emotional expressions, complicating sentiment determination. Cultural nuances further compound this complexity, influencing the sentiment encoded within the text.

Furthermore, the scarcity of NLP tools and resources makes the task even harder. Text preprocessing, a crucial step in preparing data for SA, becomes arduous due to the limited availability of essential tools like stemming and lemmatization tailored for low-resource languages. This scarcity impedes effective text processing and hinders progress in developing robust STR systems for these languages.

Muhammad et al. (2022) embarked on an extensive research endeavor aimed at constructing a comprehensive database encompassing four resource-poor African languages. Their work stands out for its innovative contributions, including the development of stopwords databases and sentiment dictionaries tailored specifically for Nigerian languages.

Kelechi et al. (2021) ventured into training a multilingual language model exclusively on low-resource African languages. Their creation, AfriBERTa, spans eleven African languages, pioneering language models for four of these languages.

Dossou et al. (2022) introduced AfroLM, a multilingual language model trained from scratch on a staggering twenty-three African languages, employing a self-active learning framework. Their research highlights AfroLM's remarkable performance surpassing several multilingual pre-trained language models, including AfriBERTa, XLM-Roberta-base, and mBERT, across various downstream natural language processing tasks such as Named Entity Recognition (NER), Text Classification (TC), and Sentiment Analysis.

2.3 Text Preprocessing

Text preprocessing is crucial in NLP fields such as STR. In both general and social text documents, noise such as typos, emojis, slang, HTML tags, spelling mistakes, and repetitive letters often appear. Improperly preprocessed text can result in incorrect analysis outcomes.

HaCohen-Kerner et al. (2019, 2020) investigated the impact of all possible combinations of six preprocessing methods on TC in three datasets. The main conclusion recommended is always to perform a systematic variety of preprocessing methods, combined with many ML methods to improve the accuracy of TC.

2.4 Text Embeddings

Text embeddings are representations of textual data in a continuous vector space, enabling algorithms to process and analyze text effectively. These embeddings capture semantic and syntactic similarities between words or documents, facilitating various NLP tasks such as sentiment analysis, document classification, and information retrieval. We used 5 basic embedding methods: TF-IDF, Doc2Vec, mUSE, LSA, LDA, and 2 improved embedding methods – BERT and Sentence Transformers with a variety of models.

TF-IDF (Term Frequency-Inverse Document Frequency) represents the importance of a word in a document relative to a collection of documents. It calculates a weight for each word based on its frequency in the document and inverse frequency across all documents. Words with high TF-IDF scores are considered more informative for distinguishing documents(Ramos, 2003).

Doc2Vec, an extension of Word2Vec, generates fixed-length vectors for entire documents. It captures semantic information by training a neural network to predict the context of words within a document. Doc2Vec assigns a unique vector to each document, enabling comparison and clustering of documents based on their content(Lau & Baldwin, 2016).

mUSE (Multilingual Universal Sentence Encoder) is a pre-trained sentence encoder

developed by Google Research. It maps variable-length text inputs into fixed-length vectors, capturing semantic similarity across multiple languages.

LSA (Latent Semantic Analysis) is a dimensionality reduction technique applied to large textual corpora. It analyzes the relationships between words and documents based on the co-occurrence of terms.

LDA (Latent Dirichlet Allocation) is a probabilistic generative model used for topic modeling. It assumes that documents are composed of multiple topics, each characterized by a distribution of words. LDA infers the underlying topic structure of a document collection and assigns a probability distribution over topics for each document.

BERT (Bidirectional Encoder Representations from Transformers), introduced by Google, employs a transformer architecture to capture bidirectional contextual information from text(Devlin et al., 2018). It consists of multiple layers of transformers, enabling it to understand the context of a word within a sentence based on both preceding and succeeding words(Chi et al., 2019).

Sentence-Transformers(ST), inspired by the success of BERT, extend its capabilities to encode entire sentences or paragraphs into fixed-length embeddings. Unlike BERT, which focuses on token-level representations, ST generates embeddings at the sentence level. These embeddings capture the contextual relationships between words within a sentence.

2.5 Task and Datasets Description

The SemRel Task 1-A is based on a collection of datasets in 9 different languages(Ousidhoum et al., 2024B). Each instance in the training, development, and test sets is a sentence pair. The instance is labeled with a score representing the degree of semantic textual relatedness between the two sentences. The scores can range from 0 (maximally unrelated) to 1 (maximally related. The size of the datasets is detailed in Appendix A. The official evaluation metric for this task is the Spearman rank correlation coefficient, which captures how well the system-predicted rankings of test instances align with human judgments.

3 System Overview

In our study, we implemented a systematic approach to enhance the learning process of our classifier. To augment the available training data, we merged the datasets of the training and development sets. This consolidation aimed to enrich the information on which our classifier is trained. Subsequently, we conducted experiments where each model was evaluated on both raw sentences and preprocessed sentences. The preprocessing steps included removing punctuation marks, numeric characters, and URL addresses, and converting text to lowercase.

At each stage of the learning process, we employed various text embedding methods to convert sentence pairs into vector pairs. These text embedding methods were pivotal in capturing the semantic relationships between sentences. Following the generation of vector pairs, we trained a regression model to learn the Semantic Textual Relatedness (STR) label between the vector pairs. The trained model was then tasked with predicting the STR level for unlabeled vector pairs present in the test set. Subsequently, we performed a comparative analysis of all results and selected the best-performing models for each language under investigation.

Furthermore, we evaluated the performance of eleven machine learning regressors to determine their efficacy in predicting the STR label. These regressors include:

Linear Regression: A basic regression model that models the relationship between independent and dependent variables linearly.

Ridge Regression: A regression model that uses L2 regularization to prevent overfitting.

Gradient Boosting Regressor: An ensemble learning technique that builds decision trees sequentially, each correcting the errors of the previous one.

AdaBoost Regressor: Another ensemble learning method that combines multiple weak learners to create a strong learner.

Support Vector Regressor (SVR): A regression algorithm that finds the hyperplane that best fits the data points while minimizing the error. SVM is a supervised learning algorithm that is used for classification and regression analysis(Cortes and Vapnik, 1995; Chang & Lin, 2011).

Stochastic Gradient Descent (SGD) Regressor: A linear model trained using stochastic gradient descent.

Bayesian Ridge Regression: A regression model that is based on the Bayes theorem (Kim et al., 2006), and assumes that features are conditionally independent given the target class, estimates the probabilities of each class and the probabilities of each feature given the class, and use it to make predictions.

Decision Tree Regressor: A regression model that partitions the data into subsets based on feature values.

Random Forest Regressor: An ensemble learning method that builds multiple decision trees and outputs the average prediction. (Breiman, 2001). It combines Breiman's "bagging" (Bootstrap aggregating) idea in Breiman (1996) and a random selection of features introduced by Ho (1995) to construct a forest of decision trees.

K Neighbors Regressor: A non-parametric regression model that predicts the output based on the average of the 'k' nearest neighbors.

MLP Regressor (Multi-layer Perceptron): A neural network model with multiple layers that learns complex data patterns. Inputs are received by the input layer, processed through the hidden layers, and produce the final output (Hassan et al., 2016).

Each regressor was evaluated based on its performance in predicting the STR label, providing insights into the effectiveness of different regression techniques in our task.

4 Experimental Setup

Our way of working was based on the train and dev datasets only. The goal was to train different models on the train dataset and select the best models according to the Spearman rank score (according to the competition requirement) on the dev dataset.

For all embedding methods(see Appendix B for details), we applied the following process. In the first step, for each language, converted the sentence pairs to vectors, using different embedding methods. Every method was checked twice – one with the original pair and one with a preprocessed pair. In total, for each language, we tested 5 classic embedding methods, 4 methods based on Sentence-Transformers, and 8 methods based on BERT.

That is, for each language different embedding methods were tested, once on raw text and once on pre-processed text, and for each of these methods we trained 11 regression models. We also trained additional BERT models for English, Spanish, Moroccan Arabic, and Algerian Arabic, so that in total we compared 3572 models (for all languages together), and at least 374 models for each language.

The following tools and information sources were utilized to apply these ML methods:

Python 3.8 programming language (Van Rossum & Drake, 2009),

Sklearn – a Python library for ML methods (Buitinck et al., 2013),

Numpy – a Python library for fast algebraic calculation (Harris et al., 2020),

Pandas – a Python library for efficient data analysis (McKinney, 2010),

TensorFlow – an open-source Python library for constructing ML-DL models (Abadi et al., 2015), and

Transformers – a Python library for natural language processing, offering pre-trained models based on transformer architecture (Wolf et al., 2020).

Hugging Face - provides a platform for data scientists to access and utilize cutting-edge models (Huggingface API, 2024).

5 Experimental Results

Table 1 presents the Spearman rank score of our models for task 1A. The table shows for each language the ideal model we received, its embedding method, whether it performed pre-processing, which regressor it used, what was the score we received in the training phase (distribution of train+dev in the ratio 20:80), what was the actual score we received after submission to the competition, and what was our position in the competition as well as what is the best result achieved. The full results can be seen in Appendix C.

It seems that vector assignment in BERT-based embedding methods was better than classical methods or Sentence Transformers library-based methods. This is probably due to the work that these models were massively trained on a lot of information, with the help of huge resources, and are therefore able to characterize vectors that optimally deliver the texts. Also, BERT models know how to characterize words with their context, and this may be a significant fact concerning the STR task.

In most languages, except Spanish and Kiryanwanda, a BERT model that is multilingual was better than a BERT model that was trained only on this or a similar language. This is a surprising figure as we were sure that a specific model would excel more reliably in this language. However, it seems that the models in low-resource languages are weaker and trained on less information compared to huge models from the multilingual genre.

Among the classical embedding methods, tf-idf seems to be the most successful method because it reaches reasonable achievements even for some of the BERT models, but is still far from the best of them.

The most prominent classifiers in the best models are the Random Forest Regressor, SVR regressor, and Bayesian regressor. They are based on classic machine learning algorithms - Random Forest, Support Vector Machine, and Naive Bayes which are recognized as classic classifiers but strong and good in many ML tasks.

Despite the well-known advantages of preprocessing methods in ML tasks, it seems that there is an overall balance between models that were quicker to preprocess their text and models that worked better on the raw text. It may be that more advanced preprocessing methods such as stemming or lemmatization will be more helpful for learning, but because In most languages it was difficult to find tools that would perform this processing of texts.

6 Conclusions and Future Research

In this paper, we describe our submissions to subtask 1-A and of SemEval-2024.

We applied 17 embedding methods to convert text into vectors, 11 supervised machine learning methods, to predict regression of STR, and did it to 9 different languages.

While our study demonstrates promising outcomes across multiple languages and embedding techniques, a comprehensive error analysis reveals nuanced challenges that warrant further investigation. We observed recurrent patterns of misclassifications, particularly in contexts characterized by linguistic ambiguities, and cultural nuances, and might be affected by the prevalence of sarcasm or irony. These findings highlight the need for robust feature representation and domain-specific adaptations to enhance the accuracy and reliability of sentiment analysis models.

Moreover, our error analysis sheds light on the impact of preprocessing strategies on model performance, revealing a delicate balance between text normalization and the preservation of linguistic subtleties. While preprocessing techniques such as stemming or lemmatization hold promise for improving model generalization, their efficacy varies across languages and datasets, necessitating careful consideration in model development pipelines. We assume that by focusing on one or two languages, we would be able to examine the specific effect of each pre-processing method, as well as focus on the unique characteristics of each language in terms of morphological structure or methods for simplifying and decomposing words, to enable better processing and better results.

There are various ideas for future research regarding the nature of Twitter messages:

(1) use mot preprocessing methods to bring the text to a more understandable shape.

(2) Trying to enrich our training dataset and tune more parameters and longer training because deep learning becomes better with more data to train and more time.

(3) Error analysis must be performed in-depth and repetitive patterns of errors, consistently incorrect classifications, etc. must be identified, to allow for the correction and improvement of the models.

The STR task is an important task that can contribute in many fields, and this study is a milestone in my acquaintance with this task and in developing the way to do it properly.

Language	Classifier	Embedding Type	Pre process	Train Score	Test Score	Rank	SemRel Best Score
Algerian Arabic	RandomForest Regressor	BERT-LaBSE2	No	0.53699	0.44273	17/24	0.68231
Amharic	SVR	BERT-LaBSE2	Yes	0.72871	0.71269	14/18	0.88863
English	SVR	BERT-bert-base-uncased	No	0.75010	0.72020	35/36	0.85958
Hausa	BayesianRidge	BERT-roberta	No	0.61895	0.54304	16/21	0.76429
Kinyarwanda	BayesianRidge	BERT-afrisenti	Yes	0.53506	0.41256	17/21	0.81691
Marathi	SVR	BERT-bert-multi	No	0.76888	0.77817	21/25	0.91086
Moroccan Arabic	SVR	tf-idf	No	0.79914	0.70112	17/23	0.86257
Spanish	BayesianRidge	BERT-robertuito	Yes	0.71538	0.66071	16/25	0.74039
Telugu	MLPRegressor	BERT-distilbert-multi	No	0.74199	0.70555	21/25	0.87336

Table 1: scores of best models for each language in task 1A.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What Makes Sentences Semantically Related? A Textual Relatedness Dataset and Empirical Study. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Leo Breiman. 1996. Bagging predictors. Machine learning 24(2), 123-140.

Leo Breiman. 2001. Random forests. Machine learning 45(1), 5-32.

Alexander Budanitsky, and Graeme Hirst. "Evaluating wordnet-based measures of lexical semantic relatedness." Computational linguistics 32.1 (2006): 13-47.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, & Gaël Varoquaux, 2013. API design for machine learning software: experiences from the scikit-learn project. In ECML PKDD Workshop: Languages for Data Mining and Machine Learning (pp. 108–122).

Chih-Chung Chang and Chih-Jen Lin, 2011. LIBSVM: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST) 2(3), 1-27.

Sun Chi, Qiu Xipeng, Xu Yige, Huang Xuanjing, 2019. "How to Fine-Tune BERT for Text Classification?." arXiv e-prints: arXiv-1905.

Corinna Cortes and Vladimir Vapnik, 1995. Support-vector networks. Machine learning 20.3 : 273-297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bonaventure Dossou, Atnafu Lambebo Tonja, Oreen Yousuf, Salomey Osei, Iyanuoluwa Shode, Oluwabusayo Olufunke Awoyomi, Chris & Chinenye Emezue, 2022. AfroLM: A Self-Active Learning-based Multilingual Pretrained Language Model for 23 African Languages. arXiv preprint arXiv:2211.03263.

Evgeniy Gabrilovich and Shaul Markovitch. "Computing semantic relatedness using Wikipedia-based explicit semantic analysis." IJcAI. Vol. 7. 2007.

Yaakov HaCohen-Kerner, Yair Yigal, and Daniel Miller. 2019. The impact of Preprocessing on Classification of Mental Disorders, in Proc. of the 19th Industrial Conference on Data Mining, (ICDM 2019), New York.

Yaakov HaCohen-Kerner, Daniel Miller, and Yair Yigal. 2020. The influence of preprocessing on text classification using a bag-of-words representation, PloS one, vol. 15, p. e0232525.

Taieb Hadj, Mohamed Ali, Torsten Zesch, and Mohamed Ben Aouicha. "A survey of semantic relatedness evaluation datasets and procedures." Artificial intelligence review 53.6 (2020): 4407-4448.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gerard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, & Travis E. Oliphant (2020). Array programming with NumPy. Nature, 585(7825), 357–362.

Ramchoun Hassan, Mohammed Amine Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil, 2016. "Multilayer Perceptron: Architecture Optimization and Training." International Journal of Interactive Multimedia and Artificial Intelligence 4, no. 1 (2016): 26+.

Tin Kam Ho. 1995. Random decision forests. Proceedings of 3rd international conference on document analysis and recognition. Vol. 1. IEEE.

Ron Keinan and Yaakov HaCohen-Kerner. "JCT at SemEval-2023 Tasks 12 A and 12B: Sentiment Analysis for Tweets Written in Low-resource African Languages using Various Machine Learning and Deep Learning Methods, Resampling,

and HyperParameter Tuning." Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023). 2023.

HuggingFace API, 2024. https://huggingface.co/docs/api-inference/index Last Access: 13/Feb/2023

Ogueji Kelechi, Yuxin Zhu, and Jimmy Lin, 2021. "Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages." Proceedings of the 1st Workshop on Multilingual Representation Learning.

Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim and Sung Hyon Myaeng, 2006. "Some Effective Techniques for Naive Bayes Text Classification," in IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 11, pp. 1457-1466, Nov. 2006, doi: 10.1109/TKDE.2006.180.

Jey Han Lau and Baldwin Timothy (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. arXiv preprint arXiv:1607.05368

Wes McKinney, 2010. Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 56 - 61).

Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani, Sebastian Ruder, Ibrahim Sa'id Ahmad, Idris Abdulmumin, Bello Shehu Bello, Monojit Choudhury, Chris Chinenye Emezue, Saheed Salahudeen Abdullahi, Anuoluwapo Aremu, Alípio Jorge, and Pavel Brazdil, 2022. NaijaSenti: A Nigerian Twitter Sentiment Corpus for Multilingual Sentiment Analysis. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 590–602, Marseille, France. European Language Resources Association.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, & Saif M. Mohammad. (2024A). SemRel2024: A Collection of Semantic Textual Relatedness Datasets for 14 Languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, & Saif M. Mohammad. (2024B). SemEval-2024 Task 1: Semantic Textual Relatedness. In *"Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)"*.

Juan Ramos (2003). "Using tf-idf to determine word relevance in document queries." Proceedings of the first instructional conference on machine learning. Vol. 242. No. 1.

Guido Van Rossum & Fred Drake, 2009. Python 3 Reference Manual. CreateSpace.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, & Alexander M. Rush, 2020. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38–45). Association for Computational Linguistics.

Appendix A - Details of the Data Sets

Language	Train Size	Dev Size	Test Size
Algerian Arabic	949	97	583
Amharic	599	95	171
English	911	249	919
Hausa	558	212	565
Kinyarwanda	435	102	222
Marathi	270	267	284
Moroccan Arabic	319	70	324
Spanish	615	139	599
Telugu	260	130	273

Appendix B – All Embedding Models

Model Name	Type	Languages
doc2vec	basic	Multilingual
mUSE		
tf-idf		
LSA		
LDA		
distiluse-base-multilingual-cased-v2	Sentence-Transformers	
paraphrase-multilingual-MiniLM-L12-v2		
LaBSE		
clip-ViT-B-32-multilingual-v1		
bert-base-multilingual-uncased	BERT	
lxyuan/distilbert-base-multilingual-cased-sentiments-student		
Davlan/afrisenti-twitter-sentiment-afroxlmr-large		
intfloat/multilingual-e5-base		
l3cube-pune/indic-sentence-similarity-sbert		
setu4993/LaBSE		
setu4993/LEALLA-large		
FacebookAI/xlm-roberta-base		
Abdou/arabert-large-algerian		Algerian + Moroccan
alger-ia/dziribert		
CAMeL-Lab/bert-base-arabic-camelbert-da-sentiment		
asafaya/bert-large-arabic		
aubmindlab/bert-base-arabert		
SI2M-Lab/DarijaBERT		
pysentimiento/robertuito-sentiment-analysis		Spanish
llange/xlm-roberta-large-spanish		
dccuchile/bert-base-spanish-wwm-uncased		
maxpe/bertin-roberta-base-spanish_sem_eval_2018_task_1		
cardiffnlp/twitter-roberta-base-sentiment-latest		English
distilbert-base-uncased-finetuned-sst-2-english		
bert-base-uncased		
roberta-large		

Appendix C - Full Results, 10 Best Models For Every Language

Language	Classifier	Embedding Type	Preprocessing	Train_Score
Algerian Arabic	RandomForestRegressor	BERT-LaBSE2	No	0.5369947229
Algerian Arabic	GradientBoostingRegressor	BERT-LaBSE2	No	0.5292887473
Algerian Arabic	RandomForestRegressor	BERT-LaBSE2	Yes	0.5253867143
Algerian Arabic	SVR	BERT-bert-multi	Yes	0.5220256197
Algerian Arabic	SVR	BERT-bert-multi	No	0.5210616209
Algerian Arabic	BayesianRidge	BERT-aubmindlab	No	0.5152289012
Algerian Arabic	BayesianRidge	BERT-aubmindlab	Yes	0.5137481244
Algerian Arabic	SVR	SenTransformers-LaBSE	No	0.5110653241
Algerian Arabic	SVR	SenTransformers-LaBSE	Yes	0.5104857707
Algerian Arabic	SVR	BERT-LaBSE2	No	0.5077027734
Amharic	SVR	BERT-LaBSE2	Yes	0.7287084049
Amharic	BayesianRidge	BERT-roberta	Yes	0.7246094157
Amharic	BayesianRidge	BERT-roberta	No	0.7204889719
Amharic	MLPRegressor	BERT-roberta	Yes	0.7080872218
Amharic	BayesianRidge	BERT-LaBSE2	Yes	0.7044388055
Amharic	SVR	BERT-LaBSE2	No	0.7023932501
Amharic	MLPRegressor	BERT-roberta	No	0.6991415451
Amharic	BayesianRidge	BERT-LaBSE2	No	0.694283367
Amharic	BayesianRidge	SenTransformers-LaBSE	Yes	0.6608416741
Amharic	Ridge	SenTransformers-LaBSE	Yes	0.6608308762
English	SVR	BERT-bert-multi	No	0.7582006981
English	SVR	BERT-bert-base-uncased	No	0.750103082
English	BayesianRidge	BERT-bert-roberta-large	No	0.741107994
English	SVR	BERT-LaBSE2	No	0.7404424659
English	BayesianRidge	BERT-bert-multi	No	0.735515388
English	Ridge	BERT-bert-roberta-large	No	0.7322067128
English	BayesianRidge	BERT-bert-roberta-large	Yes	0.731029189
English	SVR	BERT-LaBSE2	Yes	0.7307237556
English	Ridge	BERT-bert-roberta-large	Yes	0.7270267272
English	SVR	BERT-twitter-roberta	No	0.7231381043
Hausa	BayesianRidge	BERT-roberta	No	0.6189488918
Hausa	MLPRegressor	BERT-roberta	Yes	0.6104493667
Hausa	BayesianRidge	BERT-roberta	Yes	0.6077610956
Hausa	MLPRegressor	BERT-roberta	No	0.6028932623
Hausa	SVR	BERT-afrisenti	No	0.5847085719
Hausa	SVR	BERT-afrisenti	Yes	0.5814811298
Hausa	SVR	BERT-LaBSE2	Yes	0.5483401586
Hausa	BayesianRidge	BERT-afrisenti	Yes	0.5479463345
Hausa	SVR	BERT-bert-multi	No	0.5371360093
Hausa	BayesianRidge	BERT-afrisenti	No	0.5369844352

Kinyarwanda	BayesianRidge	BERT-afrisenti	Yes	0.5350585294
Kinyarwanda	SVR	BERT-afrisenti	Yes	0.5146730111
Kinyarwanda	SVR	BERT-e5-base	No	0.5136255749
Kinyarwanda	BayesianRidge	BERT-distilbert-multi	No	0.505681762
Kinyarwanda	BayesianRidge	BERT-e5-base	Yes	0.4963074713
Kinyarwanda	SGDRegressor	BERT-e5-base	Yes	0.4960261965
Kinyarwanda	MLPRegressor	BERT-roberta	No	0.4956656724
Kinyarwanda	BayesianRidge	BERT-e5-base	No	0.4947476356
Kinyarwanda	SGDRegressor	BERT-e5-base	No	0.4933142053
Kinyarwanda	GradientBoostingRegressor	BERT-distilbert-multi	No	0.4911255779
Marathi	SVR	BERT-bert-multi	No	0.768881107
Marathi	BayesianRidge	BERT-bert-multi	No	0.7688210054
Marathi	SVR	BERT-bert-multi	Yes	0.7546816577
Marathi	BayesianRidge	BERT-distilbert-multi	No	0.7532801443
Marathi	BayesianRidge	BERT-bert-multi	Yes	0.7505721435
Marathi	SVR	BERT-distilbert-multi	No	0.7467252478
Marathi	MLPRegressor	BERT-bert-multi	No	0.7440670356
Marathi	BayesianRidge	BERT-distilbert-multi	Yes	0.7415936477
Marathi	SVR	BERT-distilbert-multi	Yes	0.7414378556
Marathi	MLPRegressor	BERT-distilbert-multi	No	0.7379256945
Moroccan Arabic	SVR	tf-idf	No	0.7991443722
Moroccan Arabic	SVR	tf-idf	Yes	0.796339094
Moroccan Arabic	Ridge	SenTransformers-LaBSE	Yes	0.7787889425
Moroccan Arabic	SVR	BERT-LaBSE2	No	0.7778968174
Moroccan Arabic	BayesianRidge	SenTransformers-LaBSE	Yes	0.777541694
Moroccan Arabic	MLPRegressor	SenTransformers-LaBSE	Yes	0.772735299
Moroccan Arabic	SVR	BERT-LaBSE2	Yes	0.77245585
Moroccan Arabic	SVR	SenTransformers-LaBSE	Yes	0.7704490304
Moroccan Arabic	BayesianRidge	BERT-LaBSE2	No	0.7676106274
Moroccan Arabic	BayesianRidge	BERT-CAMeL-Lab	No	0.7665164306
Spanish	BayesianRidge	BERT-robertuito	Yes	0.7153770062
Spanish	GradientBoostingRegressor	BERT-robertuito	Yes	0.7128803162
Spanish	BayesianRidge	BERT-robertuito	No	0.7112746809
Spanish	AdaBoostRegressor	BERT-robertuito	Yes	0.6989013087
Spanish	BayesianRidge	BERT-bert-base-spanish	Yes	0.6979171296
Spanish	SGDRegressor	BERT-distilbert-multi	No	0.6971499681
Spanish	GradientBoostingRegressor	BERT-robertuito	No	0.6967140825
Spanish	BayesianRidge	BERT-distilbert-multi	Yes	0.6964474195
Spanish	SVR	BERT-LaBSE2	Yes	0.6959682585
Spanish	SGDRegressor	BERT-distilbert-multi	Yes	0.6956928668
Telugu	MLPRegressor	BERT-distilbert-multi	No	0.7419850235
Telugu	SVR	BERT-LaBSE2	Yes	0.7331294075

Telugu	SVR	BERT-bert-multi	No	0.7325062071
Telugu	BayesianRidge	BERT-distilbert-multi	No	0.7313601112
Telugu	SVR	BERT-distilbert-multi	No	0.7312296203
Telugu	BayesianRidge	BERT-bert-multi	No	0.7255846168
Telugu	SVR	BERT-bert-multi	Yes	0.722301082
Telugu	SGDRegressor	BERT-distilbert-multi	No	0.7199655333
Telugu	BayesianRidge	BERT-distilbert-multi	Yes	0.7196172815
Telugu	SVR	BERT-LaBSE2	No	0.7194733505

USMBA-NLP at SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials using Bert

Anass Fahfouh[1], **Abdessamad Benlahbib**[1], **Jamal Riffi**[1] , **Hamid Tairi**[1]
[1] LISAC Laboratory, Faculty of Sciences Dhar EL Mehraz, USMBA, Fez, Morocco
anassfahfouh@gmail.com, abdessamad.benlahbib@usmba.ac.ma ,
riffi.jamal@gmail.com, htairi@yahoo.fr

Abstract

This paper presents the application of BERT in SemEval 2024 Task 2, Safe Biomedical Natural Language Inference for Clinical Trials. The main objectives of this task were: First, to investigate the consistency of BERT in its representation of semantic phenomena necessary for complex inference in clinical NLI settings. Second, to investigate the ability of BERT to perform faithful reasoning, i.e., make correct predictions for the correct reasons. The submitted model is fine-tuned on the NLI4CT dataset, which is enhanced with a novel contrast set, using binary cross entropy loss.

1 Introduction

NLI stands for Natural Language Inference. It is a task in natural language processing (NLP) where the goal is to determine the relationship between two text segments: a premise and a hypothesis. The task typically involves classifying whether the hypothesis is entailed, contradicted, or neutral with respect to the premise.

NLI has emerged as a beacon of hope for Clinical Trial Reports (CTRs). Its ability to handle vast amounts of medical evidence could revolutionize the interpretation and retrieval of CTRs. Clinical trials stand as pillars in experimental medicine, scrutinizing the efficacy and safety of novel treatments (Avis et al., 2006). CTRs meticulously outline trial methodologies and findings, guiding the development of targeted interventions for patients. Yet, the staggering quantity of published CTRs renders manual review impractical for devising new treatment protocols (DeYoung et al., 2020).

The proposed textual entailment task aims to advance the understanding of models' behavior and enhance existing evaluation methodologies for clinical NLI. By systematically applying controlled interventions, each engineered to probe a specific semantic phenomenon in natural language and numerical inference, the study seeks to assess the robustness, consistency, and faithfulness of models in clinical settings, thereby investigating their reasoning capabilities.

In this paper, we present our findings on SemEval 2024 Task 2, Safe Biomedical Natural Language Inference for Clinical Trials (Jullien et al., 2024). The aim of our method is to assess the robustness, consistency, and faithfulness of BERT (Devlin et al., 2019), Pre-training of Deep Bidirectional Transformers for Language Understanding, on the clinical NLI. Our method follows to steps: the first step is fine-tuning BERT on the Multi-evidence Natural Language Inference for Clinical Trial Data (NLI4CT) (Jullien et al., 2023) which is enhanced with a novel contrast set. Then, the prediction step, consists on the determining the inference relation (entailment vs contradiction) between CTR - statement pairs.

The rest of the paper is structured in the following manner: Section 2 provides the main objective of the Task. Section 3 describes our system. Section 4 details the experiments. And finally, Section 5 concludes this paper.

2 Task Description

This paper focuses on the task of textual entailment within the domain of clinical trial data analysis, specifically targeting Clinical Trial Reports (CTRs). CTRs serve as comprehensive documents containing essential information regarding various aspects of clinical trials, including eligibility criteria, interventions, results, and adverse events. Automating the analysis of CTRs through natural language processing techniques can significantly facilitate researchers' understanding and decision-making processes.

The task of NLI4CT involves analyzing annotated statements and determining their inference relation with the information contained in the CTR premises. These statements, characterized by an average length of 19.5 tokens, make claims about

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 432–436

various sections of the CTRs, including:

- **Eligibility criteria**: A set of conditions for patients to be allowed to take part in the clinical trial
- **Interventions**: Information concerning the type, dosage, frequency, and duration of treatments being studied.
- **Results**: Number of participants in the trial, outcome measures, units, and the results.
- **Adverse events**: These are signs and symptoms observed in patients during the clinical trial.

The NLI4CT task presents several challenges inherent to clinical trial data analysis, including numerical and quantitative reasoning, vocabulary and syntax variations, and comprehension of complex semantic structures. To address these challenges, interventions have been implemented targeting the following aspects:

- **Numerical Reasoning**: Models' abilities to apply numerical and quantitative reasoning are specifically targeted, given the importance of such inference in clinical trial analysis.
- **Vocabulary and Syntax**: Acronyms, aliases, and syntactic patterns common in clinical texts are addressed to improve model robustness and performance.
- **Semantics**: Complex reasoning tasks involving longer premise-hypothesis pairs are intervened upon to enhance model capabilities in handling intricate semantic structures.

3 System Description

To evaluate BERT on the SemEval 2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials, we have fine-tuned BERT model on the NLI4CT dataset. we follow standard procedures for fine-tuning transformer-based models on natural language inference tasks. Here's a description of the process:

- **Data Preprocessing**: Tokenize the CTR premises and statements using the BERT tokenizer. Encode the tokenized sequences into input IDs, attention masks, and segment IDs as required by BERT.
- **Model Architecture**: Utilize the BERT architecture, which is a pre-trained transformer model. Add a classification layer on top of the BERT model to predict the entailment relation (entailment vs. contradiction) between the CTR premises and statements.
- **Fine-tuning Objective**: Fine-tune the pre-trained BERT model on the NLI4CT task using supervised learning. Minimize the binary cross-entropy loss between the predicted entailment labels and the ground truth labels.
- **Training Procedure**: Train the fine-tuned BERT model on the training data comprising CTR premises and statements along with their corresponding labels (entailment or contradiction).

4 Experimental Results

We experimented our model on on the SemEval 2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. The experiment has been conducted in Google Colab environment[1], The following libraries: Transformers - Hugging Face[2] (Wolf et al., 2020), and Tensorflow[3] were used to train and to assess the performance of the model.

4.1 Datasets

The premises within NLI4CT are sourced from 1000 publicly accessible Breast cancer Clinical Trial Reports (CTRs) in English[4]. These records are overseen by the U.S. National Library of Medicine and adhere to the HIPAA Privacy Rule. The CTRs are categorized into four sections: Eligibility criteria, Intervention, Results, Adverse Events (Jullien et al., 2023).

A team of domain experts, including organizers of clinical trials from a prominent cancer research institution, participated in annotating the data. Annotators were tasked with generating entailment statements based on two CTR premises. These entailment statements make objectively true claims about the content of the premises. Annotators could choose to create statements regarding one or both premises. Substantial statements typically involve

[1] https://colab.research.google.com/
[2] https://huggingface.co/docs/transformers/index
[3] https://tensorflow.org
[4] https://ClinicalTrials.gov

summarization, comparison, negation, relation, inclusion, superlatives, aggregation, or rephrasing, requiring an understanding of multiple aspects of the premise. Annotators then select a subset of facts from the premises to support the claims in the statement.

Subsequently, a negative rewriting technique (Chen et al., 2020) was employed, altering the previously generated entailment statements to include objectively false claims while maintaining the original sentence structure and length. This technique aims to mitigate the likelihood of stylistic or linguistic patterns favoring either entailment or contradictory statements. Annotators then extract a subset of facts from the premises that contradict the claims in the false statement.

The resulting dataset comprises 2400 annotated statements with labels, premises, and evidence. The dataset was divided into train/test/dev sets in a 70/20/10 ratio. The two classes and four sections are evenly distributed across the dataset and its partitions (Jullien et al., 2023).

4.2 Evaluation Metric

The assessment of task performance will entail multiple stages. Initially, the performance on the original NLI4CT statements without any alterations, employing the Macro F1-score for evaluation.

Subsequently, the performance will be assess on the contrast set, comprising all statements with interventions. In this evaluation, two novel metrics—faithfulness and consistency—will be utilized, with their definitions provided below.

- **Faithfulness**: quantifies how accurately a system predicts outcomes for the right reasons. Essentially, it assesses the model's capacity to adjust its predictions accurately when encountering semantic-altering interventions. To compute faithfulness for a set of N statements x_i in the contrast set C, alongside their original statements y_i and model predictions f(), Equation 1 is utilized.

$$Faithfulness = \frac{1}{N}\sum_{1}^{N} |f(y_i) - f(x_i)| \quad (1)$$

$$x_i \in C : Label(x_i) \neq Label(y_i), and f(y_i) = Label(y_i)$$

- **Consistency**: assesses how consistently a system generates identical outputs for problems that are semantically equivalent. Consequently, it gauges a system's capability to assign the same label to both original statements and contrast statements for interventions that preserve semantics. This means that even if the ultimate prediction is incorrect, the representation of the semantic phenomenon remains consistent across the statements. To calculate consistency for a set of N statements x_i in the contrast set C, along with their corresponding original statements y_i and model predictions f(), Equation 2 is employed.

$$Consistency = \frac{1}{N}\sum_{1}^{N} 1 - |f(y_i) - f(x_i)| \quad (2)$$

$$x_i \in C : Label(x_i) = Label(y_i)$$

4.3 Experimental Settings

During the fine-tuning of BERT model on the NLI4CT taining set, we set the hyper-parameters as follows: 10^{-5} as the learning rate, 30 epochs, 64 as the max sequence length, and 16 as batch size. Table 1 summarizes the hyperparameters settings of BERT base model.

Hyperparameters	Settings
Learning rate	10^{-5}
Batch size	16
Epochs	30
Max sequence length	64
Optimizer	Adam (Kingma and Ba, 2015)
Loss	Binary Cross Entropy

Table 1: Hyperparameters settings for the model in the experiments

4.4 System Performance

The reported results for the fine-tuned BERT model on the NLI4CT task are as follows:

- **Macro F1-score**: 0.62
- **Faithfulness**: 0.44
- **Consistency**: 0.54

The model achieved the 26th position in Macro F1-score, Faithfulness and Consistency among a total of 32 teams. The reported score of 0.62 in the Macro F1-score indicates that the model achieves moderate performance in accurately predicting the inference relation between CTR premises and statements. Moreover, the Faithfulness score, which is 0.44, suggests that the model struggles in making correct predictions for the right reasons. This indicates potential issues with reasoning or interpretation of the textual entailment task. On the other hand, the Consistency score, which is 0.54, indicates moderate consistency in the model's outputs for similar instances. However, there is room for improvement to achieve higher consistency.

The suboptimal performance of the fine-tuned BERT model on the NLI4CT task could be attributed to several factors: Firstly, clinical trial data, especially Clinical Trial Reports (CTRs), often contain domain-specific terminology, complex medical concepts, and nuanced language. BERT, being pre-trained on general-domain text, may struggle to comprehend and accurately reason over such specialized content. Secondly, The success of fine-tuning BERT depends on various hyperparameters, such as learning rate, batch size, and optimization algorithm. Suboptimal choices for these parameters can hinder convergence and degrade model performance. Thirdly, The interventions applied to the test set statements could introduce complexities or biases that the model is not equipped to handle, the model may struggle to generalize effectively. By addressing these factors the model performance can be improved in clinical trial data analysis tasks.

5 Conclusion

in this paper, an investigation is conducted into the utilization of BERT for NLI4CT, which underscores the complex nature of textual entailment tasks within the medical domain. The described approach tackles SemEval 2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. The model secured the 27th position among a total of 32 teams.

Despite challenges such as domain-specific terminology and nuanced semantics, our study reveals the potential for advancements in automated analysis of clinical trial reports. By recognizing the need for domain-specific approaches and leveraging the models, we pave the way for more accurate and reliable models tailored to the intricacies of medical data. Ultimately, our findings advocate for continued research and development efforts aimed at enhancing natural language processing techniques for clinical applications, thereby contributing to improved healthcare outcomes and medical decision-making processes.

References

Nancy Avis, Kevin Smith, Carol Link, Gabriel Hortobagyi, and Edgardo Rivera. 2006. Factors associated with participation in breast cancer clinical trials. *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, 24:1860–7.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jay DeYoung, Eric P. Lehman, Benjamin E. Nye, Iain James Marshall, and Byron C. Wallace. 2020. Evidence inference 2.0: More data, better models. *ArXiv*, abs/2005.04177.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Here's a breakdown of the results:

Control:

- **F1-score**: 0.6212
- **Recall**: 0.5899
- **Precision**: 0.6560

Contrast:

- **F1-score**: 0.4786
- **Recall**: 0.3655
- **Precision**: 0.6933

Faithfulness:

- **Score**: 0.4375

Consistency:

- **Score**: 0.5365

Paragraph Consistency:

- **Score**: 0.5813

Continuous Faithfulness:

- **Score**: 0.4160

Continuous Consistency:

- **Score**: 0.4080

Numerical Paragraph Consistency:

- **Score**: 0.5804

Numerical Continuous Faithfulness:

- **Score**: 0.5789

Numerical Continuous Consistency:

- **Score**: 0.6667

Definitions Consistency:

- **Score**: 0.5353

Paragraph:

- **F1-score**: 0.6293
- **Recall**: 0.5646
- **Precision**: 0.7107

Continuous:

- **F1-score**: 0.0
- **Recall**: 0.0
- **Precision**: 0.0

Numerical Paragraph:

- **F1-score**: 0.5
- **Recall**: 0.4845
- **Precision**: 0.5165

Numerical Continuous:

- **F1-score**: 0.0
- **Recall**: 0.0
- **Precision**: 0.0

Definitions:

- **F1-score**: 0.6001
- **Recall**: 0.5267
- **Precision**: 0.6973

CRCL at SemEval-2024 Task 2: Simple prompt optimizations

Clément Brutti-Mairesse
CRCL / Lyon, France
clement.bruttimairesse@lyon.unicancer.fr

Loïc Verlingue
CLB/CRCL / Lyon, France
loic.verlingue@lyon.unicancer.fr

Abstract

We present a baseline for the SemEval 2024 task 2 challenge, whose objective is to ascertain the inference relationship between pairs of clinical trial report sections and statements.

We apply prompt optimization techniques with LLM Instruct models provided as a Language Model-as-a-Service (LMaaS).

We observed, in line with recent findings, that synthetic CoT prompts significantly enhance manually crafted ones.

The source code is available at this GitHub repository github.com/ClementBM-CLB/semeval-2024.

1 Introduction

Since the introduction of large pre-trained transformer models such as GPT-3.5, released in early 2022, foundational models have begun to be utilized widely. While BERT-like models have proven to be effective in various NLP tasks such as Named Entity Recognition (Devlin et al., 2019), scaling up the number of parameters in transformer models not only enhances their capabilities but also endows them with new abilities not seen in smaller models (Zhao et al., 2023). These capabilities are particularly evident in natural language inference tasks, where the model must deduce the veracity of two given texts (Zhong et al., 2023).

LLMs, gaining popularity for their reasoning capabilities, still face trustworthiness concerns, crucial in the medical domain where decisions affect lives. Medical devices must exhibit reliability and undergo rigorous testing before they are brought to market. SemEval 2024 (Jullien et al., 2023b, 2024) focuses on assessing NLI system robustness, coherence, and accuracy, particularly LLMs prone to shortcut learning, factual discrepancies, and performance degradation from word distribution shifts (Liu et al., 2023).

Fine-tuning, while effective for task and domain adaptation, demands excessive resources in the case of large language models (LLMs). In the medical field, data is highly sensitive and protected by privacy regulations. Therefore, applying fine-tuning techniques to such sensitive data would imply that medical centers have readily available on-premise infrastructure (Sun et al., 2023). Considering these limitations, we investigate hard prompt optimization techniques such as Chain-of-Thought prompting (Wei et al., 2023). Acknowledging the in-context learning (ICL) as an indirect method of fine-tuning, we also explored in-context learning strategies (Dai et al., 2023). Among them, we were particularly inspired by MedPrompt, a promising composite prompting method applied to medical datasets, which achieved a 27% reduction in error rates on MedQA (Nori et al., 2023).

Following the SemEval 2024 task 7 (Jullien et al., 2023a), SemEval 2024 task 2 focuses on identifying the inference relationship (entailment vs. contradiction) between Clinical Trial Report (CTR) statement pairs. These statements and the supporting evidence are crafted by individuals with expertise in the clinical domain, including clinical trial organizers and research oncologists. The clinical trials information is sourced from the [1]clinicaltrials.gov website (maintained by the NIH). We have evaluated three LLM prompting methods to address this task.

2 Methods

2.1 Tasks

The challenge involves analyzing a statement alongside one or two clinical trial reports to ascertain if the statement logically follows from the information presented in the clinical trial. Typically, a statement is a concise text averaging 19.5 words and may contain one or several claims pertaining

[1]https://clinicaltrials.gov/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 437–442

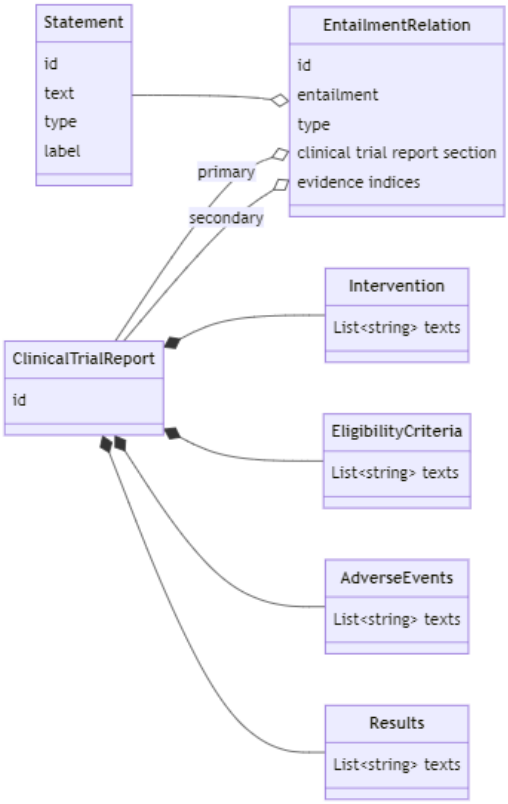


Figure 1: SemEval 2024 dataset data model

to the clinical trial. It refers to one of four sections of the clinical trial report: Adverse Events, Eligibility Criteria, Results, or Interventions. Each section represents a distinct part of the clinical trial documentation as recorded in the clinicaltrials.gov database. The text from these sections has an average length of 265 words.

For the purpose of evaluating this task, the evaluation dataset was generated to allow us to assess the reliability (faithfulness) and consistency of the inference predictions. This was achieved by paraphrasing the text to retain the same meaning, as well as by making minor alterations to the text that change the inference relationship.

2.2 Prompting

We explored three prompting optimization techniques: 1) OPRO approach, which iterates over labeled examples to determine the most effective instruction (Yang et al., 2023), 2) self-generated chain of thought (Kojima et al., 2023), 3) in-context learning (ICL) strategy by incorporating one example for one-shot prompting (Nori et al., 2023).

2.3 OPRO optimization

The OPRO technique exploits the model's capability to generate prompts based on a few exemplars and previous instructions.

In essence, the model is tasked with creating prompt instructions that are intended to solve the given problems. While this method enables the discovery of the most suitable instructions for each set, it still demands extensive resources due to its iterative optimization process. For this reason, we apply this technique to only a subset of representative examples from the development dataset.

Algorithm 1: OPRO prompt optimization

Data: N samples, M test samples and P instructions and their F1 scores
Result: P instructions
for *10 times* **do**
  Format the P instructions and N samples as a context C for the LLM
  Generate instruction with the LLM and context C
  for M *test samples* **do**
    Format the instruction and the test sample as a context
    Generate prediction with the LLM
  end
  Calculate the F1 score for the generated instruction
  Add the new instruction to the P list if its F1 score is greater than the lower instruction's score of the list
end

2.4 Zero-shot Chain-of-Thought prompt

Unlike the previous method, which constrained instructions based on the type and section of the sample, we allowed the model to generate a chain of thought reasoning using a task-agnostic meta-prompt.

The model first generate a CoT reasoning to answer the question. Then, given the previous, it is prompted to generate a conclusion and provide the final answer—whether it entails or contradicts—in a standardized json format (algorithm 2). See the figure 3 in appendix for a detailed example.

2.5 Dynamic one-shot Chain-of-Thought prompt

We hypothesized that selecting one meaningful example from a set (statement, clinical trial report)

Algorithm 2: Zero-shot Chain-of-Thought prompt

Data: N samples
Result: N predictions
for N *samples* **do**
  Format the N samples as a context $C_{reasoning}$
  Generate chain-of-thought with the LLM and the context $C_{reasoning}$
  Format the generated chain-of-thought with the sample and the formating instruction
  Generate the prediction with the LLM and the context $C_{formating}$
end

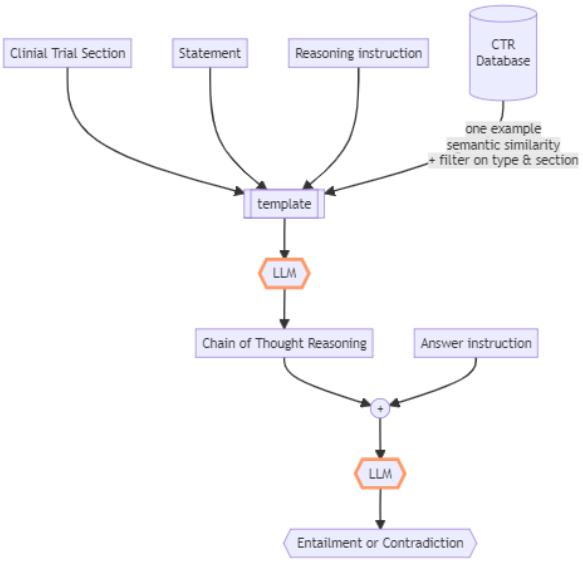


Figure 2: Dynamic one-shot prompting workflow

with a correct reasoning path could enhance the performance of the NLI system.

This experiment is divided into two tasks. First, we build a database of exemplars from the train dataset. Each sample corresponds to a statement and a clinical trial report section, along with its associated reasoning path (generated by the model) and predicted label. We filter the records where the model provides correct answers and index the embeddings of the statements into a vector database.

Next, for each test sample, we select a sample from the train dataset that is semantically close according to the squared L2 distance defined as $d = \sum \left(s_i - s_i^{train}\right)^2$. We choose the s^{train} sample with the lowest distance to the s sample that has either the same type, the same section, or both, preferably.

Algorithm 3: Vector database building

Data: N training samples
Result: Vector database of statement and reasoning paths
for N *samples* **do**
  Calculate the embeddings of the statement
  Generate prediction following the same procedure as in Algorithm 1
  If the prediction is accurate, add the embedding vector to the database
end

3 Language models

We evaluated Mixtral-8x7B-Instruct (Jiang et al., 2024), GPT3.5 (Ouyang et al., 2022), Qwen-72b-chat (Bai et al., 2023), and Mistral-7B-Instruct. For all inference tasks, except instruction generation, we did not use sampling techniques.

To calculate vector embeddings, we utilized the msmarco-bert-base-dot-v5 model, in conjunction with [2]chromadb to store the embeddings in a vector database, thereby facilitating similarity score calculations using L2 norm.

4 Evaluation metrics

Faithfulness measures the extent to which a given system arrives at the correct prediction for the correct reason. This is estimated by measuring the ability of a model to correctly change its predictions when exposed to a **semantic altering** intervention. Given N statements x_i in the contrast set (C), their respective original statements y_i, and model predictions $f()$ we compute faithfulness using Equation 1.

$$Faithfulness = \frac{1}{N}\sum_{1}^{N} |f(y_i) - f(x_i)|$$
$$x_i \in C : \text{Label}(x_i) \neq \text{Label}(y_i), \text{ and } f(y_i) = \text{Label}(y_i) \quad (1)$$

Consistency aims to measure the extent to which a given system produces the same outputs for semantically equivalent problems. Therefore, consistency is measured as the ability of a system to

[2]https://www.trychroma.com/

Model	Optimization	Base F1	Consistency	Faithfulness
Mixtral-8x7B	Zero-shot CoT	0.70	0.70	0.87
Mixtral-8x7B	Dynamic one-shot	0.60	0.71	0.89
Mixtral-8x7B	OPRO	0.59	0.65	0.81

Table 1: Prompt optimization strategies with Mixtral-8x7B-Instruct-v0.1 on the test dataset

predict the same label for original statements and contrast statements for **semantic preserving** interventions. Given N statements x_i in the contrast set (C), their respective original statements y_i, and model predictions $f()$ we compute faithfulness using Equation 2.

$$Consistency = \frac{1}{N}\sum_{1}^{N} 1 - |f(y_i) - f(x_i)| \quad (2)$$
$$x_i \in C : \text{Label}(x_i) = \text{Label}(y_i)$$

5 Results

5.1 Main results

Our team ranked sixth in faithfulness score, and we fell outside the top 10 for the baseline F1 score (0.70) and consistency (0.70). We observed that handcrafted prompts were generally less effective than optimized prompts or meta-prompts.

Prompting strategies were first tested on the dev dataset and then run on the test dataset. The results are shown in the table 1. Mixtral-8x7B-Instruct demonstrated the best quality-to-time ratio. The dynamic one-shot prompting achieved the highest Faithfulness score and Consistency score. While the best F1 score goes for the zero-shot CoT prompt approach. These results must be interpreted with caution because the model does not always return a well-formatted answer in JSON format. In cases where the answered entailment label is unknown, our approach was to prioritize the contradiction label.

Because of time limitations, we had to train and assess the prompt strategy using the development dataset, which consisted of 200 samples. We solely used the training dataset to gather examples for inputting into the vector database for the one-shot prompt strategy. The execution of the entailment task on the test dataset required 20 hours for each prompting strategy. The team's outcomes for the task are presented in table 2.

Ranking	Base F1	Base F1	Faithfulness	Consistency
1	dodoodo	0.78 (3)	0.92 (3)	0.81 (1)
2	aryopg	0.78 (5)	0.95 (2)	0.78 (2)
3	jvl	0.78 (4)	0.80 (13)	0.77 (3)
.	.	.	.	.
17	ClementBM	0.70 (18)	0.87 (6)	0.70 (17)
.	.	.	.	.

Table 2: Team ranking on the test dataset

5.2 Other evaluations

We also investigated reformulation methods, such as rephrasing negative statements, paraphrasing statements to maintain the original meaning, and rewording sections of the clinical trial report (Cheng et al., 2023), we did not observe an improvement in inference accuracy (data not shown).

We observed that applying dynamic one-shot technique (F1=0.60) obtained a 10-point drop compared to the Zero-Shot CoT (F1=0.70). We also observed that implementing preprocessing steps could improve the performance of the entailment task (such as enriching the clinical trial section with additional information, transforming negative statements into positive ones, etc.).

While experimenting with various prompt instructions to reformulate or paraphrase the statement before logical prediction on inference, we found that it didn't significantly improve performance. One detail worth mentioning is perhaps a processing step on the clinical trial report section. We observed that the model sometimes struggles to identify which paragraph of the report section matches which cohort. To address this, we explicitly added the cohort number to the subtitle of the section. All other lines of the section were concatenated without change, each separated by a newline.

6 Conclusion

By employing prompt optimization techniques with LLM Instruct models, we see the significant enhancement Zero-shot CoT prompts provide compared to manually crafted ones. This highlights the critical role of utilizing advanced techniques in LLM prompting to enhance inference tasks, particularly in domains like clinical trials.

7 Acknowledgments

This work was supported by TM2 interreg Grant from the European Regional Development Fund, TRIAL MATCH 2 (N°SYNERGIE 20023).

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen Technical Report. ArXiv:2309.16609 [cs].

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting Large Language Models via Reading Comprehension. ArXiv:2309.09530 [cs].

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why Can GPT Learn In-Context? Language Models Implicitly Perform Gradient Descent as Meta-Optimizers. ArXiv:2212.10559 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. ArXiv:2401.04088 [cs].

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. ArXiv:2205.11916 [cs].

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the Logical Reasoning Ability of ChatGPT and GPT-4. ArXiv:2304.03439 [cs].

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. 2023. Can Generalist Foundation Models Outcompete Special-Purpose Tuning? Case Study in Medicine. ArXiv:2311.16452 [cs].

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. ArXiv:2203.02155 [cs].

Qiushi Sun, Chengcheng Han, Nuo Chen, Renyu Zhu, Jingyang Gong, Xiang Li, and Ming Gao. 2023. Make Prompt-based Black-Box Tuning Colorful: Boosting Model Generalization from Three Orthogonal Perspectives. ArXiv:2305.08088 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. ArXiv:2201.11903 [cs].

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large Language Models as Optimizers. ArXiv:2309.03409 [cs].

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. ArXiv:2303.18223 [cs].

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can ChatGPT Understand Too? A Comparative Study on ChatGPT and Fine-tuned BERT. ArXiv:2302.10198 [cs].

A Prompt instructions

A.1 Zero-shot CoT prompt instruction

The following diagram illustrates with a sample from the dev dataset, how prompts are crafted. The Zero-shot CoT approach involves prompting the LLM twice.

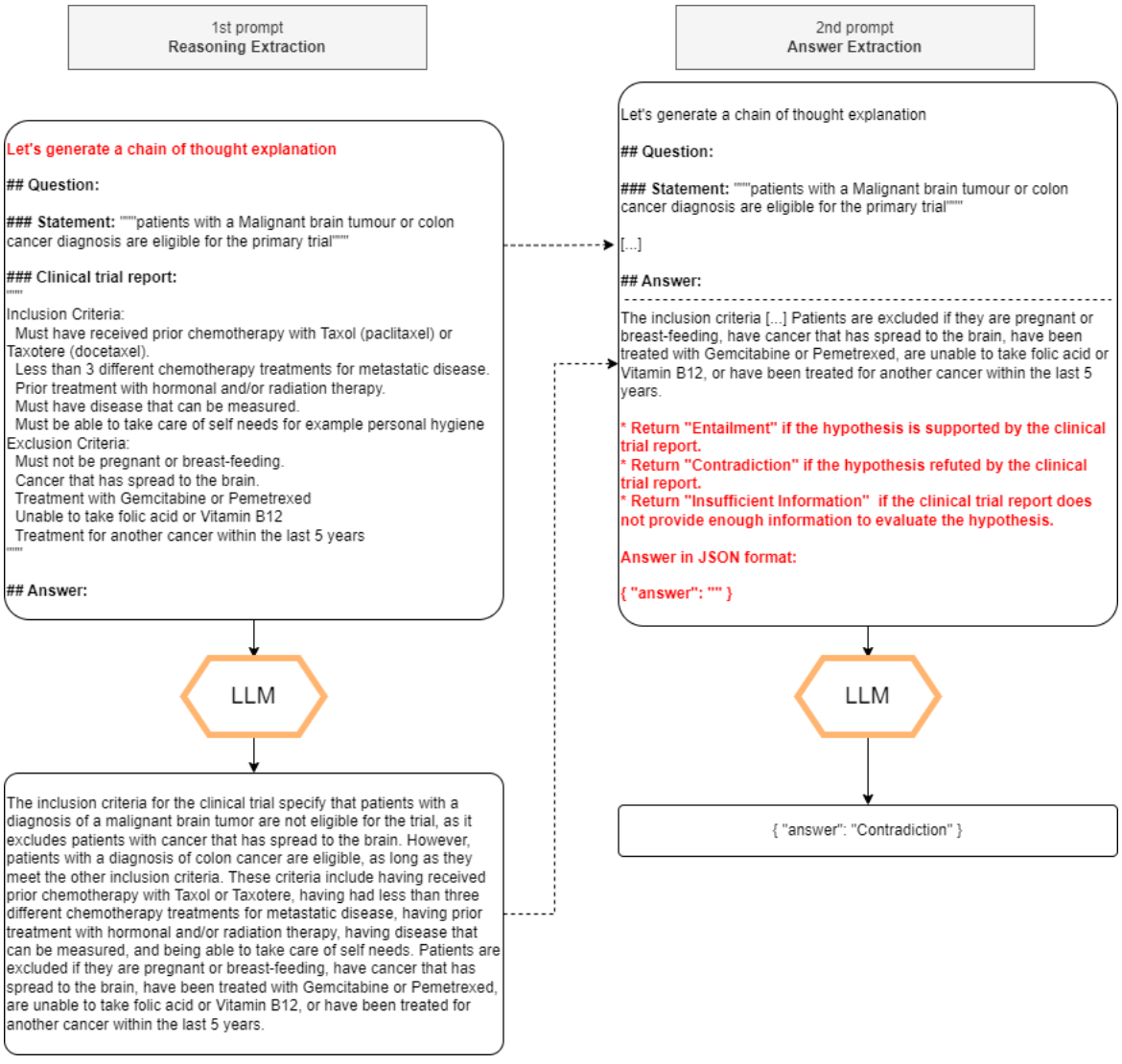


Figure 3: Zero-shot CoT prompting sample pipeline

SuteAlbastre at SemEval-2024 Task 4: Predicting Propaganda Techniques in Multilingual Memes using Joint Text and Vision Transformers

Ion-Marian Anghelina, Gabriel-Sebastian Buță and **Alexandru Enache**
University of Bucharest
Faculty of Mathematics and Computer Science
{ion.anghelina, gabriel.buta, alexandru.enache1}@s.unibuc.ro

Abstract

The main goal of this year's SemEval Task 4 is detecting the presence of persuasion techniques in various meme formats. While **Subtask 1** targets text-only posts, **Subtask 2**, subsections **a** and **b** tackle posts containing both images and captions. The first 2 subtasks consist of **multi-class** and **multi-label** classifications, in the context of a **hierarchical** taxonomy of 22 different persuasion techniques.

This paper proposes a solution for persuasion detection in both these scenarios and for various languages of the caption text. Our team's main approach consists of a Multimodal Learning Neural Network architecture, having Textual and Vision Transformers as its backbone. The models that we have experimented with include EfficientNet and ViT as visual encoders and BERT and GPT2 as textual encoders.

1 Introduction

In nowadays society, the role of social media in opinion formation is more important than ever. A fundamentally form of social media leisure, the meme has become a powerful resource which can easily be abused by various entities with political interests. The most well known platforms have a strict policy regarding misleading information, especially of political nature. However, posts containing such information are hard to automatically detect, and the administrators mostly rely on user reports.

This paper proposes an automatic detection solution for Arabic, Bulgarian, English and North Macedonian, suitable for both text-only and text-image memes. (Dimitrov et al., 2024)

Only English training data was provided for all the Subtasks, all the other languages' tasks requiring a One-Shot Learning approach.

The proposed model excels on **Subtask 2a**, scoring 2^{nd} place for all languages, besides English and also on **Subtask 2b** where it also ranked second, achieving an $F1$-score of over 0.8 in the binary setup. It struggled, however, on **Subtask 1**, especially in the case of the Subtask variants for languages without training samples, achieving an $F1$-score of about 0.2 on average.

The full results table can be found in the **Results** subsection in Table 6.

Python code for all the used models and algorithms is available in our GitHub Repository.

2 Background

2.1 Related Work

Dimitrov et al. in "Detecting propaganda techniques in memes" have also conducted their own approach of solving the **Subtask 1** and **Subtask 2b** on their own dataset in a Multimodal setup (Dimitrov et al., 2021).

Martino et al. in "A Survey on Computational Propaganda Detection" reviewed the state of the on computational propaganda detection from both the perspective of using Natural Language Processing in order to detect propaganda, as well as analysing users profiles in order to detect a propaganda networks on media platforms (Martino et al., 2020). They tackle **Subtask 1** and **Subtask 2b** as well, being, however, asked to also provide all the occurences' positions for each propaganda technique.

2.2 Input

The text information is given as a JSON which, for each meme, it has a unique **id** assigned, a **text** representing the text that is written on the meme. For the **Subtask 2a** and **Subtask 2b** we also have a **image** attribute representing the name of the image to which the previously mentioned information is corresponding.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 443–449

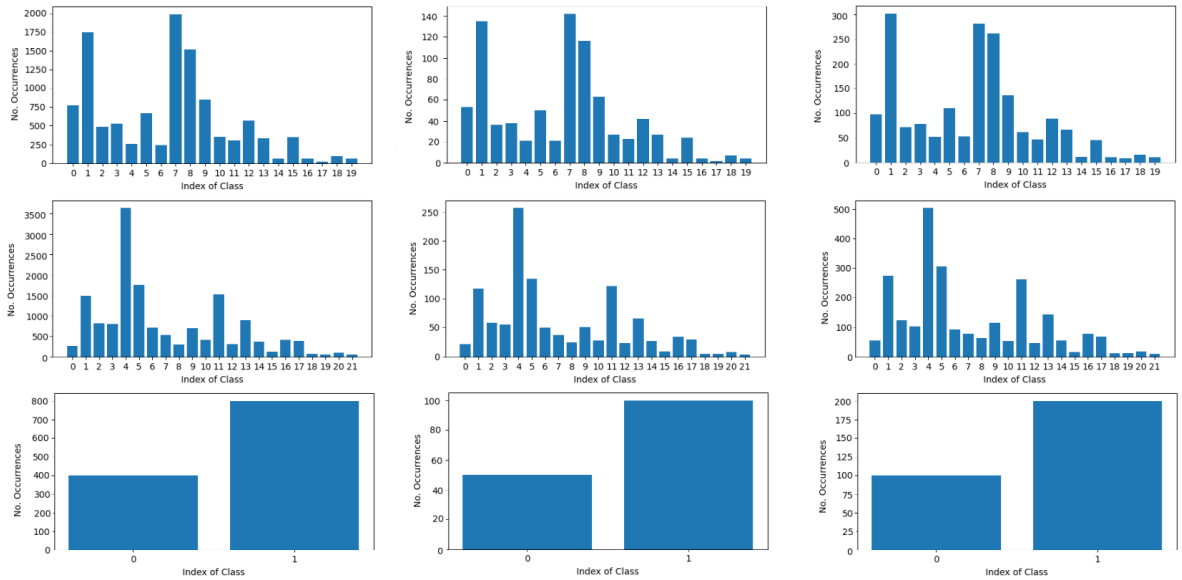


Figure 1: The class distribution for **Subtask 1**, **Subtask 2a** and respectively **Subtask 2b** (on rows) for the **train**, **validation** and respectively **dev** sets (on columns)

Below is an example extracted from the English test dataset for **Subtask 1**:

```
{
    "id": "64747",
    "text": "Just a few of the Rino's
        that need to go!!\\nRino season
        will be open soon!"
}
```

And another example from the English test dataset for **Subtask 2a**:

```
{
    "id": "79142",
    "text": "NOW ENTERING 2022",
    "image": "prop_meme_24023.png"
}
```

2.3 Output

For the **Subtask 1** and **Subtask 2a**, the output is returned in a JSON file which, for each meme, contains the unique **id** through which the photo was identified in the input, as well as a **labels** list containing the

labels corresponding to the meme. For **Subtask 2b**, instead of the labels list we will have only a **label** that is represented either by the string **propagandistic** or **non_propagandistic**. Below is an example extracted from a submission for the **Subtask 1** of the Arabic test dataset:

```
{
    "id": "00001",
    "labels": ["Black-and-white Fallacy/
        Dictatorship", "Presenting
        Irrelevant Data (Red Herring)"]
}
```

And another example from the Arabic test dataset for **Subtask 2b**:

```
{
    "id": "00007",
    "label": "non_propagandistic"
}
```

2.4 Dataset

The dataset is composed of memes with English captions present on them. For **Subtask 1** and **Subtask 2a**, we have **7000** train images, **500** validation images and **1000** dev images, while for **Subtask 2b**, we have **1200** train images, **150** validation images and **300** dev images.

From **Figure 1** we can observe that the class distribution is conserved throughout the train, validation and dev sets.

2.5 Datasets used

For the **English** task, for **Subtask 1**, **Subtask 2a** and **Subtask 2b**, we fine-tuned **limjiayi/bert-hateful-memes-expanded** (limjiayi, 2024) which is a model based on **bert-base-uncased** (Devlin et al., 2018a) which was previously fine-tuned on **HatefulMemes** (Kiela et al., 2020), **HarMemes** (Dimitrov, 2024) and **MultiOff**(Suryawanshi et al., 2020) datasets.

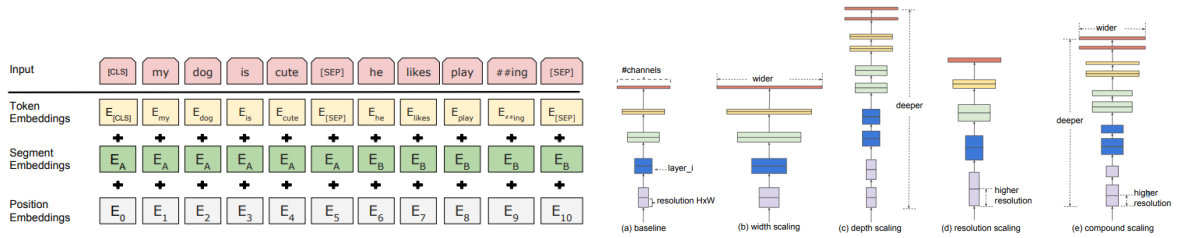


Figure 2: On the left side we have BERT Text Encoder Visual Representation (Devlin et al., 2018a) while on the right side we have the Model Scaling of an EfficientNet model (Tan and Le, 2019)

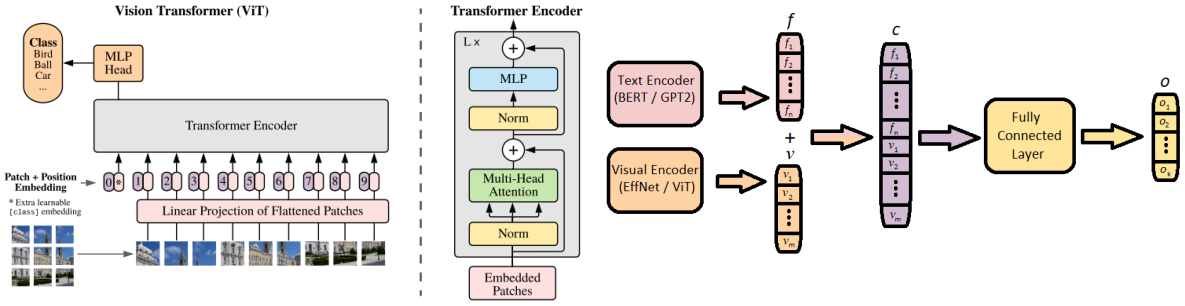


Figure 3: On the left side we have the model overview of a Vision Transformer (Radford et al., 2019) while on the right side we have our architecture of solving the proposed task

For the **Bulgarian** task, for **Subtask 1**, **Subtask 2a** and **Subtask 2b**, we used **usmiva/bert-web-bg** model (Marinova et al., 2023), an architecture pretrained from scratch BERT on Bulgarian dataset created at the Bulgarian Academy of Sciences.

For the **North Macedonian** task, for **Subtask 1**, **Subtask 2a** and **Subtask 2b**, we used **macedonizer/mk-gpt2** (Radford et al., 2019) model which is a model based on GPT2-large which was trained on English language using a causal language modeling.

For the **Arabic** task, for **Subtask 1** and **Subtask 2b**, we used **asafaya/bert-base-arabic** (Safaya et al., 2020) model which is a model based on BERT base fine-tuned on the arabic corpus. For **Subtask 2a** we used the **google-bert/bert-base-multilingual-uncased** model (Devlin et al., 2018b).

3 System Overview

3.1 Architecture Overview

For **Subtask 1**, our approach consists of using a text encoder which provides an accurate vectorial embedding of the memes captions. These representations are later used to train a **Fully Connected Neural Network**. The latter network's last layer consists of 20 **output neurons**, each representing the probability of existence of one of the 20 persuasion techniques.

For **Subtasks 2a and 2b**, the input format requires a special model branch for preprocessing the images. Similar to the **text encoder**, the **image encoder** outputs a dense feature vector representing a spacial embedding of the image. The results of the two encoders are used to train a similar Fully Connected Neural Network. In the case of **Subtask 2a**, the last layer of the neural network consist of 22 **output neurons**, each representing the probability of finding a certain persuasion technique. For subtask 2*b*, however, the last layer represents only 2 neurons, whose outputs are the probabilities of the text containing a persuasion technique or not.

While **Subtasks 1 and 2a** are treated as regression tasks with 20 and 22 output values, respectively, **Subtask 2b** is treated as a simple binary classification task.

3.2 Textual Encoders

For the text encoder branch of the model, pretrained variants of the following transformers were used:

- **BERT**(Devlin et al., 2018a) (Figure 2) is a very powerful text representation model, since it can capture large bidirectional links between words in order to build accurate word embeddings. It

relies on the self attention mechanism.

- **RoBERTa** (Liu et al., 2019) is an improved version of **BERT**, using a dynamic masking strategy during training, which is conducted over larger datasets and with a larger batch-size.

- **GPT-2** (Radford et al., 2019) is a transformer based model, pretrained on various dataset, which excels in the task of text generation based on a given prompt, but which can also be used as a backbone in any classification or regression task.

3.3 Visual Encoders

- **Vision Transformers** (Dosovitskiy et al., 2020) (Figure 3) are a class of Computer Vision models which, unlike their predecessors, do not rely on Convolutional Neural Networks as their backbone, but utilize the **Transformer** architecture, adapted from **NLP** tasks. It bears a resemblance to **BERT**, the main difference being that patches of the image are used instead of words. It was pretrained on the **ImageNet_21k** dataset. (Ridnik et al., 2021)

- **EfficientNet** (Tan and Le, 2019) (Figure 2) is a class of Convolutional Neural Network models which achieve high performance on image classification tasks. They mainly rely on automatically scaling the depth and width the network with respect to fixed parameters regarding the dataset to obtain the best possible results.

3.4 Predicting the answer

For **Subtask 1**, we denote $h_{i,j}$, the output of the **last hidden layer** of the textual encoder model the j^{th} token of the i^{th} sample.

Every sample in the training set has a fixed dimension of t tokens, and each token is embedded into a 768 dimensional vector. The fully connected neural network has a hidden layer of size d.

We first flatten $h_{i,j}$, obtaining f, a dense vector of embeddings for each word.

If we denote by (W, b) the tensor of weights between the transformer output layer and the output layer of the network, and the biases of the output layer respectively, the output of the output layer will be represented by:

$$W \cdot \tanh(f) + b$$

The output layer has 20 neurons. The result for each of them is computed by applying the $Sigmoid$ activation function, which outputs a probability.

Thus, the final output of the model is:

$$Sigmoid(W \cdot \tanh(f) + b)$$

For **Subtasks 2a and 2b**, the dense feature vector f is defined similarly, representing the textual extracted features.

We define similarly v, the vector of features extracted by the **Visual Encoder** and c, the vector obtained by concatenating f and v.

$$c = fv$$

For **Subtask 2a**, the output of the model is very similar to the one of **Subtask 1**, the main difference being that the output layer now has 22 neurons. (Figure 3)

The output of this model is:

$$Sigmoid(W \cdot tanh(c) + b)$$

For **Subtask 2b**, the output of the model is a probability distribution, the 2 classes of the binary classification being dependent of each other. We can obtain such an output using the $Softmax$ activation function.

The output is, thus:

$$Softmax(W \cdot tanh(c) + b)$$

Where W and b are defined similarly to **Subtask 1**.

The outputs of the model for **Subtasks 2a** and **2b** are, then, compared to a threshold, which represents the lower limit of the probability for a persuasion technique to be considered used.

3.5 Transfer Learning for the Visual Encoder

During the competition, the teams were not provided training datasets in **Bulgarian, North Macedonian or Arabic**. An interesting approach we tried was using the **English** labeled data for training our model and, after that, using the resulting **Vision Encoder** component of this model in conjunction with an adequate pretrained **Textual Encoder** for the desired language.

Using this approach, we made use of the image information in the dataset, which is agnostic to the language of the meme.

3.6 Adaptive Thresholding

Since we are treating **Subtask 1** and **Subtask 2a** as regression tasks (our models output a number between 0 and 1 for each persuasion technique), we needed to choose a threshold which determines the minimum value such that a persuasion technique should be included in the answer or not. In our final submissions, we have chosen **0.25** as the threshold for all the techniques. We have determined by experimenting on the dev dataset that this threshold maximizes the Hierarchical F1 score by creating a balance between the Hierarchical Precision and the Hierarchical Recall.

One idea which we tested only after the competition was to have a separate threshold for each persuasion technique. The method used to determine this was by using the validation dataset in order to find the threshold which maximizes the F1 score for that technique. An efficient way to find the best threshold is to sort the predictions and start with a threshold of 0. This would mean that all the samples are considered true, so $TP = sum(GT)$, $FP = N - sum(GT)$ and $FN = 0$, where N is the number of samples and GT is the array of ground truths. We can calculate the $F1$ score using the formula $2TP/(2TP+FP+FN)$. Now we go though all the prediction in order and assume that the current threshold is the value of the prediction. This means that the current sample i is now considered false, so if $GT_i = 1$, then $TP = TP - 1$ and $FN = FN + 1$, else $FP = FP - 1$. This allows to compute the current $F1$ score in $O(1)$.

Also, we have found that sometimes this algorithm would find very low values, so we have limited the thresholds to 0.2 as the minimum value.

Table 1: Results after the competition using adaptive thresholding on the test dataset

Subtask	F1-Score
1 English	0.647
2a **English**	**0.695**

This method performed worse for **Subtask 1** compared to our final submission (0.657), but performed better for **Subtask 2a**, where it improved our final result of 0.684 to 0.695. (Table 1)

We also tested a smaller threshold of 0.2 for the **Arabic Subtask 2a** which managed to get 1^{st} place on the final standings with a score of 0.585.

4 Experimental Setup

For the training of models we have used the data splits in the following manner. Before the dev gold labels were available, we have used the validation dataset in order to find the best hyperparameters and after that added the validation dataset to the training data. After the dev gold labels were published, in order to submit on the test dataset, we have also trained on the dev dataset. This maximized the number of training samples available.

The preprocessing techniques used for the images are:

- Resizing the image to 224×224 pixel size to match pretraining size for the Vision Transformer
- Scaling the values on all color chanels with a standard distribution $\mathcal{N}(0.5, 0.5^2)$ to increase numerical stability and facilitate learning

The preprocessing techniques used for the texts are:

- Lowercasing all characters, necessary especially in the context of memes which do not follow a casing norm
- Tokenizing the texts into representative tokens using the pretrained tokenizer associated with the used model
- Padding or truncating the texts to a standard dimension of 128 for the transformer model

All the external libraries (and their versions) used for the setup are listed in our GitHub Repository in the requirements.txt file.

The experiments conducted can be seen in tables 2, 3 and 4.

Epochs ALL	Epochs FC	Thresh	hP	hR	hF_1
5	5	0.25	0.62	0.60	0.612
3	3	0.25	0.63	0.66	**0.646**

Table 2: Experiments for **Subtask 1** on the dev dataset

Epochs ALL	Epochs FC	Thresh	hP	hR	hF_1
3	3	0.5	0.77	0.52	0.62
3	3	0.25	0.69	0.66	0.680
3	0	0.3	0.70	0.66	0.684
3	0	0.25	0.67	0.70	**0.687**

Table 3: Experiments for **Subtask 2a** on the dev dataset

Epochs ALL	Epochs FC	F_1 $macro$	F_1 $micro$
3	0	0.735	0.760
3	3	0.754	0.773
5	5	**0.819**	0.846

Table 4: Experiments for **Subtask 2b** on the dev dataset

Subtask	Visual Encoder	Text Encoder	Epochs ALL	Epochs FC
1 English	ViT	BERT	3	3
2a English	ViT	BERT	3	3
2b English	ViT	BERT	5	5
1 Bulgarian	ViT	BERT	3	3
2a Bulgarian	ViT	BERT	3	0
2b Bulgarian	ViT	BERT	0	5
1 N. Macedonian	ViT	GPT2	3	3
2a N. Macedonian	ViT	GPT2	3	0
2b N. Macedonian	ViT	GPT2	5	5
1 Arabic	ViT	BERT	3	3
2a Arabic	ViT	BERT	0	3
2b Arabic	ViT	BERT	0	5

Table 5: The configurations of our final submissions on the test dataset

In Table 5, by "Epochs ALL" and "Epochs FC" we refer to the epochs for which the whole model is trained, respectively to the epochs where only the last fully connected layers are trained and the encoders are frozen. In the case of the non-english subtasks, the text encoder was always frozen during the training. In the cases where "Epochs ALL" appears as 0, we have experimented with taking the trained encoder from the corresponding english subtask and freezing it in the training process.

The first task we approached was **Subtask 2b**. We first used varieties of the **EfficientNet** model for the Visual Encoder, later switching to **ViT** encoding thanks to its higher performance.

The metrics used for **Subtask 1** and **Subtask 2a** are Hierarchical Precission (hP), Hierarchical Recall (hR) and Hierarchical F1 score (hF_1). In the multi-label settings, we define $\hat{C}_i$ as the set of ground truth classes and all their ancestors and $\hat{C}'_i$ as the set of predicted classes and all their ancestors (Kiritchenko et al., 2006). Thus the metrics are represented by :

$$hP = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i|}{\sum_i |\hat{C}'_i|} \quad hR = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i|}{\sum_i |\hat{C}_i|}$$

$$hF_1 = \frac{2 \cdot hP \cdot hR}{hP + hR}$$

For **Subtask 2b**, the metrics used are the classic F_1 $macro$ and F_1 $micro$.

5 Results

Subtask	F1-Score	Place
1 English	0.657	9th
2a **English**	**0.684**	**5th**
2b **English**	**0.809**	**2nd**
1 Bulgarian	0.235	19th
2a **Bulgarian**	**0.610**	**2nd**
2b Bulgarian	0.594	7th
1 N. Macedonian	0.203	19th
2a **N. Macedonian**	**0.575**	**2nd**
2b N. Macedonian	0.177	14th
1 Arabic	0.234	16th
2a **Arabic**	**0.516**	**2nd**
2b Arabic	0.500	10th

Table 6: Results during the competition on the test dataset

As it can be seen from the results tables 5 and 6, the best approach, in terms of fully training our model, rather than only fine-tuning the final classification layer, was varied. The latter approach being more suitable, especially for **Subtasks 2a** and **2b** in the case of languages without additional training data.

Our model managed to outperform the Competition Baseline for **Subtasks 2a** and **2b** for all the provided languages.

In the case of **Subtask 1**, our model only outperformed the Competition Baseline for the English language dataset.

6 Conclusion

Our model accurately creates dense embeddings for both the memes and their captions, managing to make use of their most prominent features in computing the result. The performance peaks on **Subtask 2a**. The Transformer components used in the proposed architectures are capable of learning from vast datasets with low risk of overfitting. Thus, one way of improving the solution would be the use of additional datasets related to the subject, in various languages. Last but not least, our future work will revolve around decision making on the **Hierarchical DAG**, further making use of relationship between different labels.

7 Acknowledgments

We would like to especially thank our colleague and collaborator Adrian-Ștefan Miclăuș (University of Bucharest), for continuously reviewing our work and offering valuable advice.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dimitar Dimitrov. 2024. bert-hateful-memes-expanded. https://github.com/di-dimitrov/harmeme.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Detecting propaganda techniques in memes. *arXiv preprint arXiv:2109.08013*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *CoRR*, abs/2005.04790.

Svetlana Kiritchenko, Richard Nock, and Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. volume 4013, pages 395–406.

limjiayi. 2024. bert-hateful-memes-expanded. https://huggingface.co/limjiayi/bert-hateful-memes-expanded.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Iva Marinova, Kiril Simov, and Petya Osenova. 2023. Transformer-based language models for bulgarian. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 712–720.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020. A survey on computational propaganda detection. *arXiv preprint arXiv:2007.08024*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (multioff) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020)*. Association for Computational Linguistics.

Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

RFBES at SemEval-2024 Task 8: Investigating Syntactic and Semantic Features for Distinguishing AI-Generated and Human-Written Texts

Mohammad Heydari Rad[*1], **Farhan Farsi**[*1], **Shayan Bali**[*1],
Romina Etezadi[2] **and Mehrnoush Shamsfard**[3]

[1] Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
[2] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
[3] Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran

mhrad81@aut.ac.ir, farhan1379@aut.ac.ir, shayanbali@aut.ac.ir, retezadi@uottawa.ca, m-shams@sbu.ac.ir

Abstract

Nowadays, the usage of Large Language Models (LLMs) has increased, and LLMs have been used to generate texts in different languages and for different tasks. Additionally, due to the participation of remarkable companies such as Google and OpenAI, LLMs are now more accessible, and people can easily use them. However, an important issue is how we can detect AI-generated texts from human-written ones. In this article, we have investigated the problem of AI-generated text detection from two different aspects: semantics and syntax. Finally, we presented an AI model that can distinguish AI-generated texts from human-written ones with high accuracy on both multilingual and monolingual tasks using the M4 dataset. According to our results, using a semantic approach would be more helpful for detection. However, there is a lot of room for improvement in the syntactic approach, and it would be a good approach for future work.

1 Introduction

Large Language Models (LLMs) are widely used. They are easily accessible, and people can use them by passing their queries to chatbots to generate their desired texts for several purposes and, more importantly, in different languages. Although LLMs have their own advantages and simplify the text generation process for humans, they have increased concerns about the misuse of this technology for adversarial purposes such as generating hallucinations, misinformation, disinformation, and fake news. Furthermore, improper use of LLMs can cause disruption in students' learning process.

This issue has led to research on detecting AI-generated texts versus human-written ones, and a number of articles have investigated this classification task. However, the main focus of the presented works has mainly been on the semantic aspect of this text classification task. In this article, we have investigated this issue using two different approaches to consider both the semantic and syntactic aspects of texts. To achieve this aspiration, we have developed two different models for both syntactic and semantic-based analysis to apply them to both multilingual and monolingual datasets. In this way, we used the M4 article's dataset (Wang et al., 2023) for both multilingual and monolingual tasks.

For the syntax analysis of this task, we have developed an Attention-based Long Short-Term Memory(LSTM) model to cover the complexities related to long sentences and the relationship between different parts of a sentence, and regarding the semantic analysis of this task, we have developed a transformer-based model.

According to the results, our systems have performed better than M4's provided baseline in the multilingual task, and in the monolingual task, our results are really close to M4's provided baseline. In the end, we have provided our results and compared the results of both these models with each other and with previous works in this area.

2 Background

Today, due to the remarkable advancements in Natural Language Processing (NLP), models like ChatGPT, GPT-3 (Brown et al., 2020), Gemini (formerly known as Bard) (Team et al., 2023), and others have reached a point where they can generate texts that closely resemble human writing. Consequently, the task of identifying texts generated by AI has become increasingly important. This task holds significant value across various domains, including content moderation, plagiarism detection, and ensuring transparency in AI-generated content. The approaches for this task can be categorized into three categories: (1) Deep Learning-based Detection, (2) Statistical Discrepancy Detection, and (3) Watermark-based Detection. Deep learning-based models can be formulated as a classification task where the input is a text that can be generated by either a human or an AI. The model is trained with labeled data, where each text is assigned a label indicating whether it was generated by AI or by a human. This allows the model to learn patterns and features that can accurately classify texts based on their origin. These methods are susceptible to adversarial attacks, which can manipulate the input text to deceive the model's classification. However, deep learning-based models generally demonstrate good performance on the training data distribution (Guo et al., 2023). Statistical Discrepancy Detection methods first learn the patterns of AI-generated and human-written texts separately. Then, they iden-

[*]equal contribution

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 450–454

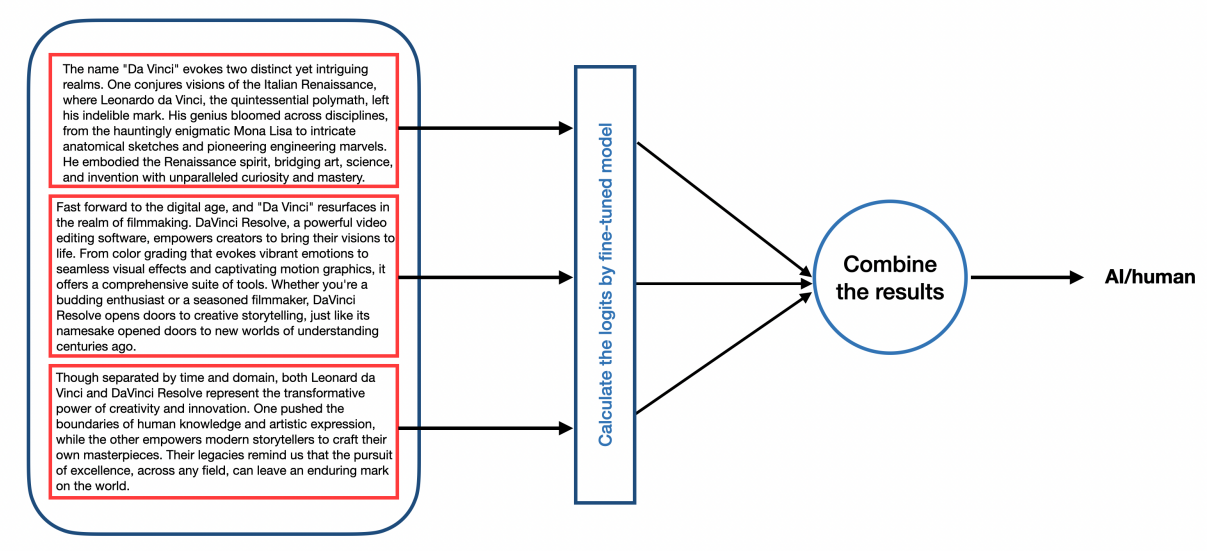


Figure 1: The input text is divided into meaningful units, and the probability of each segment based on their logits is assessed using a fine-tuned XLM-RoBERTa model; the combined evidence leads to a definitive classification.

tify statistical discrepancies between these patterns to distinguish between the two. By analyzing various linguistic features, such as word frequencies, sentence structures, or syntactic patterns, these methods can detect differences that arise from the distinct nature of AI-generated and human-written texts. Some tools like GPTZero (Mitchell et al., 2023) use perplexity (how well a language model predicts the next word based on the previous ones) and burstiness (variations in sentence length) to assess whether the text is AI-generated or human-written. The idea of watermarking initially emerged from the field of computer vision and has since been applied to NLP (Wu et al., 2023). This method involves embedding a hidden "watermark" during the text generation process with the objective of identifying text generated by a specific language model. In the context of black-box language models, (Yang et al., 2023) utilize this watermarking method to detect and identify text generated by such models.

In the SemEval2024 Task 8, (Wang et al., 2024), our attention was directed towards the multilingual and monolingual tracks of Subtask A. This subtask was designed for binary classification to distinguish between texts written by humans and those generated by machines. In terms of the dataset employed for this task, we utilized the multilingual dataset provided in the M4 article. The human-written texts in this dataset were collected from diverse sources spanning different domains. These sources include Wikipedia (March 2022 version), WikiHow, Reddit (ELI5), arXiv, and PeerRead for English. For Chinese, the texts were sourced from Baike and Web question answering (QA). Additionally, texts from news sources were included for Urdu and Indonesian, while for Russian, texts were obtained from RuATD. For Arabic, the texts were collected from Arabic Wikipedia. For the monolingual section, we have used the English corpora. In this dataset, AI-generated texts leverage multilingual LLMs such as ChatGPT, textdavinci-003, LLaMa, FlanT5, Cohere, Dolly-v2, and BLOOMz. These models undertake diverse tasks, including creating Wikipedia articles from titles and abstracts (from arXiv), generating peer reviews from titles and abstracts (PeerRead), answering questions from platforms like Reddit and Baike/Web QA, and composing news briefs based on the title. This dataset contains 122k human–machine parallel data in total, with 101k for English, 9k for Chinese, 9k for Russian, 9k for Urdu, 9k for Indonesian, and 9k for Arabic, respectively (Wang et al., 2023).

For our experiment, we used the English corpora of this dataset in the monolingual track. For the multilingual track, we utilized the whole dataset, which contains human-written and AI-generated texts from six different languages: English, Arabic, Chinese, Indonesian, Russian, and Urdu. As it is evident, our model's input is text documents, and its output is a single label that specifies whether the given text is human-written or AI-generated.

3 Method

To classify texts as either AI-generated or human-written, we have examined two crucial aspects: semantics and syntax. Our analysis of these aspects, which is detailed below, aims to identify distinctive features.

3.1 Semantic Approach

In our exploration of the semantic aspects of texts, we centered our analysis on two key elements: the vocabulary choices employed by the writer and the manner in which words are structured and combined. To achieve this, we leveraged transformers, which utilize word embeddings to capture meaning and positional encoding to account for word order and sentence structure. However,

a significant challenge lies in differentiating between AI-generated and human-written texts, especially in longer pieces, as AI models become increasingly adept at mimicking human writing styles. To address this challenge, particularly in longer texts, we proposed and adopted the strategy of splitting the text into smaller paragraphs. This allowed for a more focused and detailed analysis of each individual segment, potentially revealing subtle semantic nuances that might be overlooked in a holistic approach.

Our methodology is implemented in three distinct stages: (1) text segmentation, where the input text is divided into meaningful units; (2) probability calculation, where the likelihood of each segment being AI-generated or human-written is assessed; and (3) final prediction, where the combined evidence leads to a definitive classification. A visual representation of our approach is shown in Figure 1.

In the first stage, the input text was segmented into smaller units by splitting it at points where specific markers, such as exclamation marks, question marks, and periods, appeared within paragraphs. Additionally, during this stage, a dataset was generated to fine-tune our model.

For the second stage, we fine-tuned an XLM-RoBERTa model (Conneau et al., 2019) on the aforementioned dataset. Due to limited resources and constraints, the model was trained for only three epochs with a learning rate of `10^-8`. After the text was segmented and the model was fine-tuned, we proceeded to analyze the characteristics of each segment and calculate the probability of it being AI-generated or human-written according to their logits. To determine the final results, we employed several methods to combine the results, which are outlined below:

- **Soft voting prediction:** In this approach, we calculate the average probability of segments; if the calculated average is higher than the threshold (0.95), we conclude that the text is AI-generated.
- **Hard voting prediction:** In this approach, we calculate the probability of each segment; if it is higher than the threshold, we consider it to be AI-generated, and if more than half of the segments are considered AI-generated, then we conclude that the text is also AI-generated.
- **Weighted soft voting:** This approach is like soft voting, but we give weight to each segment; the weight of each segment is based on the number of words it contains.

We use this 0.95 threshold because of the small number of epochs the model has been trained on data.

3.2 Syntactic Approach

Another aspect that we examined was the syntactic properties of the texts. To analyze these properties, we utilized the Part-of-Speech (POS) labels associated with the words in the text. Our goal was to classify AI-generated and human-written texts based on their POS patterns.

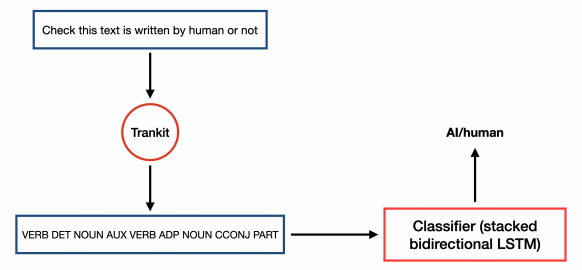


Figure 2: The bidirectional LSTM model predicts using part-of-speech labels associated with the words in the text assessed by Trankit.

To create a dataset for training a model on this aspect, we employed Trankit(Van Nguyen et al., 2021), which provided us with the Universal POS (UPOS) tokens. We integrated these UPOS tokens to form sequences of UPOS strings. In this approach, the focus is on identifying patterns within the sequences rather than the specific meaning of the tokens.

Given the challenge of working with long sequences, we opted to use an LSTM model. To handle the complexity of the task, we employed stacked LSTM layers. Additionally, to enhance the model's performance, we utilized bidirectional LSTMs, which consider the context from both directions of the sequence. In order to further improve the model's ability to capture important syntactic patterns and dependencies, we incorporated an attention layer into our LSTM model. The attention mechanism allows the model to focus on specific parts of the input sequence when making predictions, assigning different weights to different elements in the sequence. As illustrated in Figure 2, the UPOS strings produced by trankit are fed into our stacked bidirectional LSTM for classification.

By using LSTM models instead of transformer models, we aimed to prevent the potential effects of semantic meaning from overshadowing the syntactic patterns. This choice allowed us to place emphasis on the structural aspects of the texts and better isolate the syntactic features for classification purposes.

By combining the LSTM architecture with an attention layer, we aimed to enhance the model's ability to capture and utilize the important syntactic patterns in the text, ultimately improving the accuracy and effectiveness of the classification process. However, the results indicate that there is no specific difference between AI-generated and human-written texts in terms of their UPOS (Universal Part-of-Speech) patterns. We attained an accuracy of 49.75% and an F1 score of 33% for both the micro and macro averages. Therefore, we only considered the semantic aspects and overlooked the syntactic aspects.

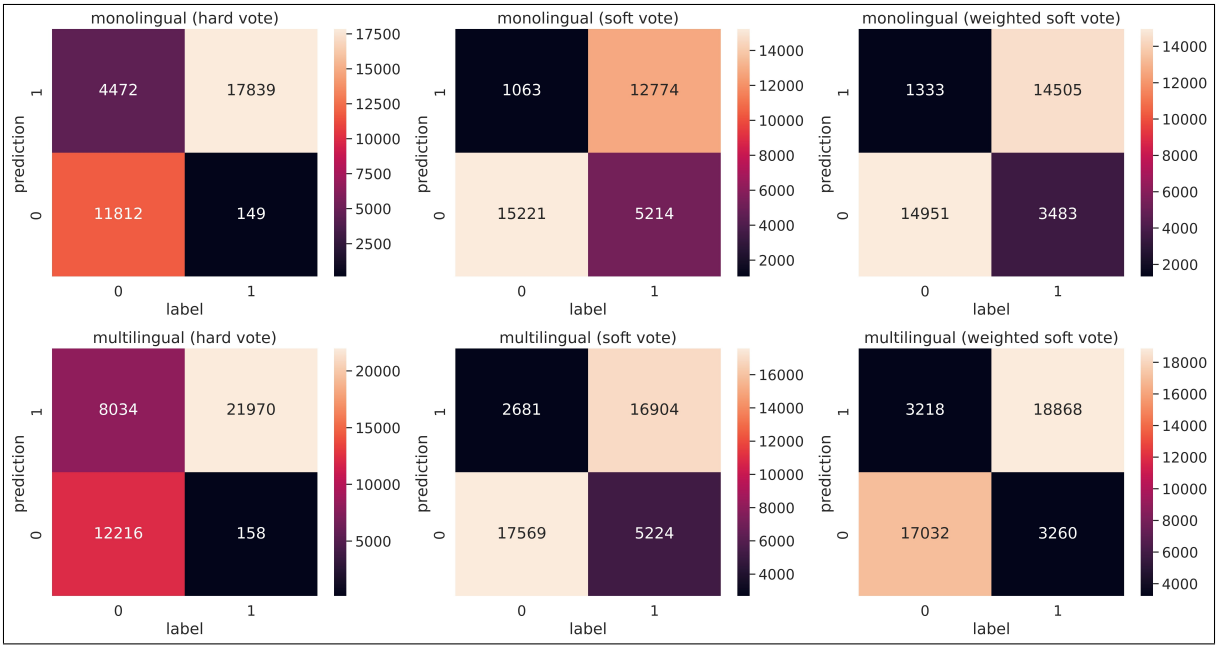


Figure 3: Confusion matrices of our model for test datasets on monolingual and multilingual tracks

4 Result

The results for our model are shown in Table 1. Our model achieved 0.847 and 0.859 accuracy for multilingual and monolingual test datasets, respectively, by training only on the multilingual training dataset.

As can be seen in the confusion matrix plots in Figure 3, the weighted soft vote approach performs better than the soft vote approach, which, in turn, outperforms the hard vote approach. Among these, the hard vote approach exhibits a higher false positive error rate compared to the other two. The soft vote approach, while having a slightly lower false positive error rate, incurs a significantly higher false negative error rate than the weighted soft vote approach.

By taking a look at mispredicted samples, we realized that the model is weak in predicting formal texts, like texts about history, law, or academic topics.

metric	multilingual	monolingual
accuracy	0.847	0.859
precision	0.854	0.916
recall	0.853	0.806
f1	0.853	0.858
false positive rate	0.159	0.082
false negative rate	0.147	0.194

Table 1: Performance of our classifier according to official metrics

5 Conclusion

In this study, we proposed a system to distinguish between human-generated and AI-generated texts. Our approach considered both semantic and syntactic aspects. For the semantic analysis, we focused on smaller text segments instead of the entire document, as we believed that AI models could produce similarly coherent long texts as humans. The results confirmed our assumption.

Our syntactic analysis, which employed a basic model to categorize texts based on their grammatical patterns using UPOS tags, revealed no significant differences in UPOS tag distribution between AI-generated and human-written texts. However, the analysis of word order identified distinct patterns in the semantic approach. This finding suggests that relying solely on UPOS tags for differentiation may be insufficient.

In conclusion, our proposed system demonstrated superior performance compared to the official baseline, achieving a 3.9% improvement in the multilingual subtask. These results emphasize the significance of considering texts in smaller segments rather than analyzing them as a whole. Moreover, our discoveries suggest that focusing solely on grammar, as indicated by their UPOS tags, may not sufficiently distinguish between AI-generated and human-written texts. Therefore, for future research efforts, it might be beneficial to utilize Graph Neural Networks (GNNs) to examine the grammatical connections among words. This involves representing word embeddings as nodes and their grammatical relationships as edges based on their constituency parsing or dependency parsing trees.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child,

Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. *arXiv preprint arXiv:2101.03289*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902v1*.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F Wong, and Lidia S Chao. 2023. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724*.

Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. 2023. Watermarking text generated by black-box language models. *arXiv preprint arXiv:2305.08883*.

BAMBAS at SemEval-2024 Task 4: How far can we get without looking at hierarchies?

Arthur B. Vasconcelos[1], **Luiz Matos**[1], **Eduardo Corrêa Gonçalves**[2], **Eduardo Bezerra**[3], **Aline Paes**[1] **and Alexandre Plastino**[1]
[1] Institute of Computing, Universidade Federal Fluminense, Niterói, RJ, Brazil
[2] National School of Statistical Sciences (ENCE/IBGE), RJ, Brazil
[3] Federal Center for Technological Education of Rio de Janeiro (CEFET/RJ), RJ, Brazil
{*athurbittencourt,lfmatosmelo*}*@id.uff.br*,
eduardo.correa@ibge.gov.br, *ebezerra@cefet-rj.br*, {*alinepaes,plastino*}*@ic.uff.br*

Abstract

This paper describes the BAMBAS team's participation in SemEval-2024 Task 4 Subtask 1, which focused on the multilabel classification of persuasion techniques in the textual content of Internet memes. We explored a lightweight approach that does not consider the hierarchy of labels. First, we get the text embeddings leveraging the multilingual tweets-based language model, Bernice. Next, we use those embeddings to train a separate binary classifier for each label, adopting independent oversampling strategies in each model in a binary-relevance style. We tested our approach over the English dataset, exceeding the baseline by 21 percentage points, while ranking in 23th in terms of hierarchical F1 and 11st in terms of hierarchical recall.

1 Introduction

In the multilabel classification problem (MLC), each instance may belong to zero, one, or multiple class labels. The goal is to learn a system to infer the correct labels of previously unseen instances (Gonçalves et al., 2018; Mylonas et al., 2023). MLC has several real-world applications, ranging from text categorization (Shimura et al., 2018) to protein and gene function prediction (Cerri et al., 2012). This work addresses a critical novel application of MLC: detecting persuasion techniques in memes, considering only their textual content, a subtask of SemEval-2024 task4[1].

The Merriam-Webster dictionary[2] defines *meme* as "an amusing or interesting item (such as a captioned picture or video) or genre of items that is spread widely online, especially through social media". Nonetheless, and unfortunately, in recent years, memes have been used not only to amuse people but also as a tool for disseminating disinformation in political campaigns (Renee, 2018; DeCook, 2018). Malicious actors embed sophisticated propaganda and persuasion techniques within these memes, employing psychological and rhetorical strategies. This manipulation extends to the memes' textual and visual components (Dimitrov et al., 2021).

Like other computational propaganda (Da San Martino et al., 2020), memes significantly influence public opinion. Their effectiveness stems from their widespread reach, potentially impacting millions of internet users globally. Additionally, memes are often not perceived as propaganda by these users, primarily because they do not mirror the appearance of conventional political advertisements (Nieubuurt, 2021).

As an effort to address this problem, SemEval-2024 Task 4 (Dimitrov et al., 2024) promoted a challenge in which competitors should develop algorithms to identify the use of persuasion techniques in memes, considering only their textual content (Subtask 1) or text and image together (Subtasks 2a and 2b). In this paper, we describe our approach to addressing Subtask 1. For this subtask, the shared-task organizers made available a collection of 8,500 texts in English extracted from real Internet memes (7,000 for training and the remaining divided into validation and dev sets). Each text may be assigned to a set of labels that indicate the persuasion techniques present in it[3]. There are a set of 20 possible labels organized in a hierarchy – thus, we have a hierarchical multilabel classification problem (Cerri et al., 2012). Some texts can have no label assigned, indicating they do not correspond to propaganda.

The shared task aimed to produce the best model according to the hierarchical-F1 metric. Test collec-

[1] https://propaganda.math.unipd.it/semeval2024task4/

[2] https://www.merriam-webster.com/wordplay/meme-word-origins-history

[3] Labels definitions are presented at https://propaganda.math.unipd.it/semeval2024task4/definitions.html

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 455–462

tions in four different languages were made available: English, Bulgarian, North Macedonian, and Arabic. Our team (BAMBAS) participated in the English challenge along with 31 other teams. We explored a lightweight approach based on three components. The first component is a language model from which we extract embedding features leveraging the [CLS] token. The second component is a binary relevance-based strategy to train 20 separate binary classifiers (one for each existing label) (Boutell et al., 2004). Our central inquiry focused on assessing the extent to which such a lightweight model that does not engage with the intricacies of hierarchical structures could be effective. The third core component handles the inherent imbalance of multilabel hierarchical problems by employing an independent oversampling strategy (Chawla et al., 2002; Menardi and Torelli, 2012) to reduce the imbalance between negative and positive examples present in each binary problem derived.

The hierarchical-F1 score of our submitted solution exceeded the baseline by 21 percentage points. In the hierarchical F1-based rank, we were the 23th out of 31 teams. However, when considering the hierarchical recall, we were ranked as 11st [4].

The rest of the paper is organized as follows. Section 2 briefly overviews MLC concepts relevant to this paper. Section 3 details our proposed system. In Sections 4 and 5, we present the experimental methodology and report the results, respectively. Finally, Section 6 brings the conclusion and future research directions.

2 Background

Over the last 20 years, MLC has been one of the most active research topics in machine learning (Mylonas et al., 2023). Among the several methods for multilabel learning in the literature (Bogatinovski et al., 2022; Prabhu et al., 2018), Binary Relevance (BR) (Boutell et al., 2004) stands out as one of the most prominent methods. This approach decomposes the multilabel problem into q binary problems, where q is the number of labels. Then, one binary classifier is independently trained for each label. The labels of new instances are predicted by combining the outputs of each classifier.

The BR method offers several key advantages. Firstly, its simplicity and intuitiveness make it highly accessible. Additionally, BR models can predict label sets not present in the training set, owing to their composition as a series of independent binary classifiers. Most crucially, BR has consistently exhibited high prediction accuracy values across various domains. In a recent extensive experimental comparison (Bogatinovski et al., 2022) involving 26 methods across 42 datasets, models utilizing BR outperformed all models trained with other different transformation strategies.

Nonetheless, the BR method suffers from three major drawbacks. First, it ignores the possible correlations among labels (Zhang et al., 2018). Second, BR has high training and prediction times for problems in which the number of labels is huge (tens of thousands to millions) (Prabhu et al., 2018). Third, its predictive performance is affected by class imbalance, which occurs when the number of examples relevant to each label is much inferior to the number of irrelevant ones (Mylonas et al., 2023; Zhang et al., 2018).

We consider that the first two drawbacks are not crucial for addressing SemEval-2024 Task 4 Subtask 1, as the number of labels in the problem is not large ($q = 20$) and there is no strong correlation between any pair of labels in the training set. More specifically, we found that the highest Pearson correlation value is 0.13 – between labels "Glittering generalities (Virtue)" and "Flag-waving". On the other hand, we consider that the issue of class imbalance needs to be taken into account as the imbalance ratio (ratio of negative to positive examples) is 47.38 on average in the training set, and the maximum value reaches 332.33 for the label "Obfuscation, Intentional vagueness, Confusion". Our approach is detailed in the next section.

3 System overview

The shared task proposed in SemEval-2024 Task 4 comprises an output of one or more labels – in case the meme is a propaganda – disposed in a hierarchical taxonomy of persuasion techniques. The root of such hierarchy is naturally labeled *persuasion*, while the second level has three possible branches: *ethos*, *pathos*, *logos*. While *ethos* and *logos* conduct to labels in a third level, *pathos* branch connects directly to the persuasion techniques – the leaves of the tree. This way, the final output can be one or more paths from the root to some leaf.

Handling such a hierarchical structure directly is quite challenging in machine learning. The algorithms should accurately predict multiple outputs

[4]Our code and experiments are available at https://github.com/MeLLL-UFF/bambas

while respecting the labels' hierarchical relationships. However, errors can propagate down the hierarchy. Moreover, some paths have very few instances, adding another layer of complexity to the problem: data sparsity and imbalance.

Therefore, our primary solution to the problem was to investigate how far an algorithm that disregards the hierarchy could go. Additionally, we also decided not to handle the multiple labels directly. However, employ the binary-relevance approach and consider a component to handle imbalance by adding synthetic instances with SMOTE (Chawla et al., 2002) and RandomOverSampler (Leevy et al., 2018), for each binary problem.

Algorithm 1 depicts the training procedure and Algorithm 2 the inference. Our method hinges on three core components. The first one creates the features from the meme textual content, leveraging a pre-trained language model (line 3 in Algorithm 1). The second component addresses class imbalance by creating synthetic instances (line 10 in Algorithm 1). The third component trains independent binary classifiers (line 12 in Algorithm 1), employing the binary-relevance strategy. During the inference phase, each label classifier undergoes evaluation, and the instance is assigned all the positive classifications predicted by each classifier.

Algorithm 1 Top-level Training Algorithm of BAMBAS team participation in SemEval-2024 Task4

1: $feats \leftarrow \emptyset, pos \leftarrow \emptyset, neg \leftarrow \emptyset, c_{labels} \leftarrow \emptyset$
2: **for** $meme \in dataset$ **do**
3: $\quad emb \leftarrow ptlm(meme.text)$
4: $\quad feats.append(\text{CLS token from } emb)$
5: $\quad$ **for** $label \in meme.labels$ **do**
6: $\quad\quad pos[label] \leftarrow pos[label] \cup meme.index$
7: $\quad$ **for** $label \notin meme.labels$ **do**
8: $\quad\quad neg[label] \leftarrow neg[label] \cup meme.index$
9: **for** $label \in labels$ **do**
10: $\quad aug\_pos[label], aug\_neg[label] \leftarrow oversampler(feats, pos[label], neg[label], rate)$
11: **for** $label \in labels$ **do**
12: $\quad c_{label} \leftarrow train\_classifier(feats, aug\_pos[label], aug\_neg[label])$
13: $\quad c_{labels} \leftarrow c_{labels} \cup c_{label}$
14: **return** c_{labels}

Algorithm 2 Top-level Inference Algorithm of BAMBAS team

1: $emb \leftarrow ptlm(meme\_text)$
2: $plabels \leftarrow \emptyset$
3: **for** $label \in labels$ **do**
4: $\quad plabel \leftarrow c_{label}(emb)$
5: $\quad$ **if** $plabel = True$ **then**
6: $\quad\quad plabels \leftarrow plabels \cup label$
7: **return** $plabels$

3.1 Extracting embedding from a pre-trained language model

In line with our straightforward premise, we have implemented a feature-based strategy that utilizes embeddings from pre-trained language models (PTLMs). The textual content of each meme is processed through the PTLM, allowing our system to capture the numeric feature vector from the [CLS] token. This method effectively harnesses the power of PTLMs to distill complex language information into a manageable form for further training our classifiers.

Our selection choice for PTLMs includes a writing free-style multilingual model, namely, XLM-RoBERTa (Conneau et al., 2020) and two informal writing style models, one monolingual (BERTweet (Nguyen et al., 2020)) and one multilingual (Bernice (DeLucia et al., 2022)).

XLM-RoBERTa is a multilingual adaptation of the RoBERTa model, pre-trained on a 2.5TB of data across 100 languages. RoBERTa itself is a transformers model trained on large raw text corpora. The key training method used is Masked Language Modeling (MLM), where 15% of the words in a sentence are masked, and the model predicts these masked words, learning a bidirectional representation of the sentence. BERTweet is a monolingual model trained from 850M Tweets. It has the same architecture as BERT-base but was trained using the RoBERTa pre-training procedure. Bernice is a multilingual RoBERTa language model trained from 2.5 billion tweets.

3.2 Training classifiers for each class

In our approach, we implemented the binary relevance strategy to train a suite of independent classifiers, each tailored to manage a binary prediction, of whether a meme belongs to a specific label. Our model comprises independent binary classifiers, each aligned to a distinct persuasion technique. Un-

der this strategy, a classifier corresponding to a label k is trained using a targeted approach: instances labeled with k are treated as positive examples, while all other instances are considered negative. This selective process ensures that each classifier becomes specialized in precisely identifying its respective label.

For instance, consider a meme m_j tagged with three labels (k_1, k_2, k_3). This meme serves as a positive training example for the classifiers c_{k_1}, c_{k_2}, and c_{k_3}, contributing to their ability to recognize these specific labels. Conversely, another meme m_z tagged with label (k_1) not only acts as a positive example for training the classifier c_{k_1} but also serves as a negative instance for c_{k_2} and c_{k_3}. This dual role of memes in the training process, as both positive and negative examples depending on their label associations, underscores each classifier's nuanced and specialized training within our binary relevance framework.

During the inference phase, each meme is processed through all the classifiers in our system. If a particular classifier predicts the meme as a positive instance, the corresponding label is assigned to the meme. By the time this processing is finished, the input meme accumulates a set of labels, each representing a positive prediction from the respective binary classifiers. This method ensures that the meme is comprehensively evaluated for all potential labels.

3.3 Creating synthetic instances

Considering the inherently imbalanced nature typical of multilabel hierarchical tasks (Mylonas et al., 2023; Zhang et al., 2018), we address this challenge by oversampling the dataset with synthetic instances. This approach is designed to equalize the number of examples for each binary classifier, thereby mitigating the imbalance issue. Our system generates synthetic examples independently for each binary classifier.

We leveraged two strategies: a simple random oversampler and the widely-used SMOTE (Synthetic Minority Oversampling Technique) (Chawla et al., 2002). SMOTE operates by identifying examples that are closely situated in the feature space. It then generates a line connecting these examples and creates a new and synthetic sample at a point along this line.

More precisely, for each classifier c_{k_i}, the process starts by selecting a random example from the minority class. Next, it identifies n nearest neighbors for this example. From these neighbors, one is randomly chosen. Subsequently, a synthetic example is crafted at a randomly determined point between the chosen neighbor and the original example in the feature space.

4 Experimental setup

Our solution was implemented using HuggingFace (Wolf et al., 2020)[5] and scikit-learn (Pedregosa et al., 2011)[6] libraries. The experiments were conducted on an NVidia DGX-1, using a single Tesla P100 GPU with 16GB of VRAM. We conducted a step-by-step analysis to reach the final modeling decisions. The intermediate results, necessary to decide the components of our final solution, are reported with the validation set. The final solution was trained with the training set and we report the results of the English dev and test set. We proceed this way because the dev set could only be measured with the submission page before the release of its gold labels.

All results are reported with the competition's evaluation metric, a hierarchical variant of the F1-score (Kiritchenko et al., 2006). The metric considers the classification taxonomy, rewarding a full score for exact leaves prediction, and rewarding a partial score for ancestor predictions. The closer the predicted ancestor is to the correct labels, the higher the partial score. Additionally, we report the hierarchical variants of precision and recall.

The first analysis consists of defining the PTLM to extract the embeddings. We did not employ oversampling during this phase and applied a binary-relevance model using logistic regression. The PTLM and logistic regression hyperparameters were left as default. The meme textual content is presented to the PTLM without any pre-processing. Next, we explore 6 other classifiers besides logistic regression: decision tree, extra tree, extra trees, KNN, random forest, and ridge classifier. The last analysis focused on selecting the best oversampling strategy. We experimented with SMOTE and a random oversampling strategy, both implemented in the imbalanced-learn library[7]. All the results so far included 20 binary classifiers, each associated with a persuasion technique in the leaves of the tree. Then, we investigate a final possibility of including some internal nodes related to the classes that

[5] https://huggingface.co/
[6] https://scikit-learn.org/stable/
[7] https://imbalanced-learn.org/stable/

were worst classified. The best model from those analyses was submitted to the competition.

5 Results

Table 1 depicts the results achieved by each pre-trained language model mentioned in Section 3.1 considering logistic regression and no oversampling strategy. Bernice achieves the best overall hierarchical-F1 (H-F1) results. We hypothesize that it was trained with a large set of informal texts from tweets, presenting a writing style close to those found in memes. Then, we select Bernice for the next analyses and to submit our final solution.

PTLM	H-F1	H-Prec.	H-Rec.
Bernice	**0.4996**	0.6246	**0.4163**
BERTweet	0.4334	0.7202	0.3100
XLM-RoBERTa	0.2928	**0.7410**	0.1825

Table 1: Validation results for choosing the PTLM

The next analysis concerns the method used as the base classifier of the binary relevance strategy. Table 2 depicts the results of the binary relevance when executed with each classifier. Logistic regression conducted to the best H-F1 score. Because of that, we proceed to the final analysis with it.

Classifier	H-F1	H-Prec.	H-Rec.
Log. Regression	**0.4996**	0.6246	**0.4163**
Decision Tree	0.3993	0.3856	0.4141
Extra Tree	0.3885	0.3826	0.3946
Extra Trees	0.1024	0.6831	0.0554
KNN	0.4252	0.5824	0.3348
Random Forest	0.1561	**0.8091**	0.0864
Ridge	0.4027	0.7388	0.2768

Table 2: Validation results of distinct Classifiers

Next, we explore our third core component, the oversampling technique. Table 3 shows the results of running the random oversampler and SMOTE with 50/50 rate for oversampling, and also a hybrid version which combines the best oversampler for each binary classifier, using different oversampling rates: 0.1 to 1.0 with step of 0.1.

Finally, we included additional classifiers to some internal nodes of the hierarchy. Such an extension includes only the internal nodes corresponding to the least accurately classified leaf nodes. These nodes are "Ad Hominem", "Distraction" and "Logos". Table 4 shows the validation set results without and with the addition of those internal nodes. Recall improved with the combined strategy, while precision remained nearly identical.

Strategy	H-F1	H-Prec.	H-Rec.
No Oversampling	0.4996	**0.6246**	0.4163
50/50 SMOTE	0.5456	0.4510	0.6904
50/50 Random	0.5383	0.4395	**0.6944**
Combination	**0.5487**	0.4783	0.6435

Table 3: Validation Results of Oversampling Strategies

Classifier	H-F1	H-Prec.	H-Rec.
W/O int. nodes	0.5487	**0.4783**	0.6435
+ int. nodes	**0.5548**	0.4782	**0.6607**

Table 4: Validation results with some internal nodes

Given the preceding results, we selected that approach for the final submission. Table 5 shows the final results achieved by the solution we submitted to SemEval-2024 Task4 platform. In the first line, we highlight the results achieved on the dev set while the second line shows (in bold) the test set result.

Set	H-F1	H-Prec.	H-Rec.
dev	0.5759	0.5046	0.6707
test	**0.5767**	**0.5012**	**0.6788**

Table 5: Final results for the official submission on both dev and test sets

6 Conclusion

This paper addresses the SemEval-2024 competition with a lightweight solution to investigate how a model that neglects the hierarchy would behave in a hierarchical task. Our solution uses a tweets-based PTLM as a feature extractor, generates synthetic data to account for imbalance, and employs a binary relevance strategy to handle multiple labels. Our next step is to investigate training a structured output classifier that predicts the paths in the hierarchy. Moreover, given that oversampling strategies enhanced the performance of most of the classes, we plan to design other strategies explicitly tailored to the style of memes.

Acknowledgments

This research was financed by CNPq (National Council for Scientific and Technological Development), under grants 311275/2020-6

(Aline Paes) and 315750/2021-9 (Alexandre Plastino) and FAPERJ - *Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro*, processes SEI-260003/000614/2023, SEI-260003/002930/2024 (Aline Paes) and E-26/201.139/2022 (Alexandre Plastino).

References

Jasmin Bogatinovski, Ljupčo Todorovski, Sašo Džeroski, and Dragi Kocev. 2022. Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications*, 203:117215.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. 2004. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771.

Ricardo Cerri, Rodrigo C. Barros, and Andre C. P. L. F. de Carvalho. 2012. A genetic algorithm for hierarchical multi-label classification. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, page 250–255, New York, NY, USA. Association for Computing Machinery.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020. A survey on computational propaganda detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, pages 4826–4832.

Julia R. DeCook. 2018. Memes and symbolic violence: #proudboys and the use of memes for propaganda and the construction of collective identity. *Learning, Media and Technology*, 43(4):485–504.

Alexandra DeLucia, Shijie Wu, Aaron Mueller, Carlos Aguirre, Philip Resnik, and Mark Dredze. 2022. Bernice: A multilingual pre-trained encoder for Twitter. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6191–6205, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Detecting propaganda techniques in memes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6603–6617, Online. Association for Computational Linguistics.

Eduardo Corrêa Gonçalves, Alex A. Freitas, and Alexandre Plastino. 2018. A survey of genetic algorithms for multi-label classification. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*, pages 395–406. Springer.

Joffrey Leevy, Taghi Khoshgoftaar, Richard Bauder, and Naeem Seliya. 2018. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(42).

Giovanna Menardi and Nicola Torelli. 2012. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28:92–122.

Nikolaos Mylonas, Ioannis Mollas, Bin Liu, Yannis Manolopoulos, and Grigorios Tsoumakas. 2023. On the persistence of multilabel learning, its recent trends, and its open issues. *IEEE Intelligent Systems*, 38(2):28–31.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Joshua Troy Nieubuurt. 2021. Internet memes: leaflet propaganda of the digital age. *Frontiers in Communication*, 5:547065.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 993–1002, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Diresta Renee. 2018. Computational propaganda: If you make it trend, you make it true. *The Yale Review*, 106(4):12–29.

Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2018. HFT-CNN: Learning hierarchical category structure for multi-label short text categorization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 811–816, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Min-Ling Zhang, Yukun Li, Xu-Ying Liu, and Xin Geng. 2018. Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 12:191 – 202.

A Validation set classification results per-label

Due to the imbalanced nature of the explored problem, we further investigate the classification results on a per-label basis. Table 6 describes the results for each label in the validation set. We include the internal nodes in the hierarchy alongside all leaves. To calculate scores for the internal nodes, predictions of any of their children are considered as correct node predictions. The best-performing label was "Appeal to Authority", which achieved the highest F1 score. The internal nodes "Logos" and "Ad Hominem" follows in second and third place, respectively. Also, most of the worst performing labels have scarce examples on the datasets, like "Vagueness, Confusion" and "Straw Man".

B After competition deadline results: multilabel classifiers

The solution presented in the paper relaxes the multiple labels per example setting and trains independent binary classifiers for each class. We additionally explored an alternative setup that leverages a multi-layer perceptron (MLP) classifier with a multilabel classification layer. We trained two classifiers in this way: the first follows the previous feature-based approach to train a multilabel feedforward (FF) MLP; the second adds a multilabel classification layer on top of the PTLM and fine-tunes all its weights. As before, the PTLM is Bernice.

The feature-based approach classifier includes a single 768-dimension hidden layer with scikit-learn default parameters. The fine-tuning approach runs for five epochs, with a learning rate of $3.9e-5$ and weight decay of $1e-3$, all selected with the validation set. Both approaches did not involve oversampling, and the classifiers were trained with the union of the train and validation sets and evaluated on the dev set during training.

Tables 7 and 8 depict the results for the dev and test sets. The results show the superior performance of the fine-tuning approach, with test set H-F1 score higher than our official competition's submission. Also, the standalone FF classifier achieved F1 above average, indicating that a dedicated oversampling strategy for the multilabel approach is a promising research avenue to explore further in the future.

Label	F1	Prec.	Rec.
Appeal to Authority	**0.7194**	0.6578	**0.7936**
Logos (internal node)	0.6965	**0.7307**	0.6653
Ad Hominem (internal node)	0.6751	0.6986	0.6530
Smears	0.5460	0.4971	0.6056
Loaded Language	0.5202	0.4782	0.5703
Name calling/Labeling	0.5119	0.4776	0.5517
Flag-Waving	0.4615	0.3870	0.5714
Black-and-White/Dictatorship	0.4000	0.3472	0.4716
Repetition	0.3859	0.3235	0.4782
Slogans	0.3650	0.2873	0.5000
Bandwagon	0.3000	0.2307	0.4285
Glittering Generalities (Virtue)	0.2807	0.2051	0.4444
Thought-Terminating cliché	0.2635	0.1868	0.4473
Exaggeration/Minimisation	0.2597	0.2000	0.0000
Appeal to Fear/Prejudice	0.2474	0.1714	0.4444
Distraction (internal node)	0.2439	0.3846	0.1785
Doubt	0.2222	0.1578	0.3750
Causal Oversimplification	0.1666	0.1282	0.2380
Whataboutism	0.0338	0.0263	0.0476
Presenting Irrelevant Data	0.0000	0.0000	0.0000
Reductio ad Hitlerum	0.0000	0.0000	0.0000
Vagueness, Confusion	0.0000	0.0000	0.0000
Straw Man	0.0000	0.0000	0.0000

Table 6: Validation set results for each task label, sorted by descending F1

Classifier	H-F1	H-Prec.	H-Rec.
Bernice$_{emb}$ $\rightarrow$ FF	0.5063	0.7257	0.3887
Bernice$_{class}$	**0.5724**	**0.7431**	**0.4655**

Table 7: Dev set results for the multilabel classifiers

Classifier	H-F1	H-Prec.	H-Rec.
Bernice$_{emb}$ $\rightarrow$ FF	0.5044	0.7177	0.3889
Bernice$_{class}$	**0.5840**	**0.7594**	**0.4744**

Table 8: English test set results for the multilabel classifiers

Team QUST at SemEval-2024 Task 8: A Comprehensive Study of Monolingual and Multilingual Approaches for Detecting AI-generated Text

Xiaoman Xu, Xiangrun Li, Taihang Wang, Jianxiang Tian, Ye Jiang
College of Information Science and Technology
Qingdao University of Science and Technology, China

Abstract

This paper presents the participation of team QUST in Task 8 SemEval 2024. We first performed data augmentation and cleaning on the dataset to enhance model training efficiency and accuracy. In the monolingual task, we evaluated traditional deep-learning methods, multiscale positive-unlabeled framework (MPU), fine-tuning, adapters and ensemble methods. Then, we selected the top-performing models based on their accuracy from the monolingual models and evaluated them in subtasks A and B. The final model construction employed a stacking ensemble that combined fine-tuning with MPU. Our system achieved 8th (scored 8th in terms of accuracy, officially ranked 13th) place in the official test set in multilingual settings of subtask A. We release our system code at:https://github.com/warmth27/SemEval2024_QUST

1 Introduction

Large language models (LLMs) enable quick, coherent responses and content creation but also raise ethical concerns about misinformation and academic integrity (Wang et al., 2023). To differentiate between machine-generated and human-created content, previous study (Guo and Yu, 2023) has been extensively discussed in industry and academic works.

Semeval 2024's Task 8 (Wang et al., 2024a) encourages the participants to develop an automatic system for detecting AI-generated text by leveraging an extended version of the M4 dataset (Wang et al., 2023, 2024b). We engaged in subtasks A and B, during which we encountered the challenges of overcoming linguistic differences, data scarcity, and inadequate cross-lingual generalization capabilities. Furthermore, existing multilingual models are less discussed in detecting AI-generated text, compared to monolingual ones, which further exacerbates the difficulty in model selection.

Meanwhile, we found that the multilingual dataset in subtask A contains data from both monolingual data and subtask B, as shown in Figure 1. To enhance the diversity and scale of the text dataset, we performed back-translation on the multilingual training set to increase the volume of monolingual data and conducted data cleaning to improve data quality.

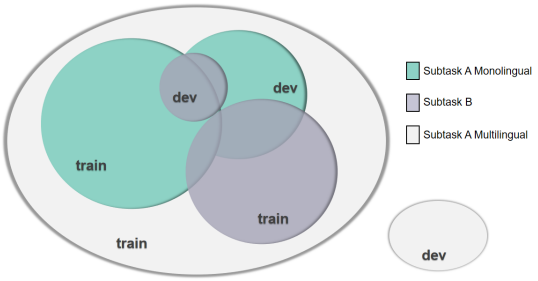


Figure 1: The data distribution in subtask A and B.

In our approach to subtasks A and B, we initially applied deep learning methods for a swift assessment in subtask A and proceeded to fine-tune multiple pre-trained language models (PLMs), inspired by recent studies highlighting the efficacy of fine-tuning methods in text classification. However, the training cost of the fine-tuning method is relatively high. We further utilize the Adapter (Hu et al., 2021) to parameter efficiency fine-tune (PEFT) (Hu et al., 2021) the model while preserving its performance. We also noted that despite the reduced training time, the performance of adapter models was not consistently stable, showing variability across experiments. Given the critical importance of model performance, decided not to utilize adapters in the testing phase.

To enhance model performance and generalizability, we adopted a stacking-based ensemble learning method, utilizing the logits from the top two performing models as inputs for a linear layer to generate final predictions. Finally, our experimental results on the test set show that integrating

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 463–470

data augmentation and ensemble learning significantly improves model efficacy in task-specific settings.

2 System description

2.1 MPU framework

Recent research on machine-generated text recognition has evolved into treating it as a binary classification problem, with the latest advancements including the Multiscale Positive-Unlabeled (MPU) (**?**) training framework. This approach introduces a length-sensitive MPU loss combined with abstract recurrent models and a text multi-scale module, significantly enhancing detection performance for short texts.

Upon analyzing the text lengths in official datasets, As shown in table 2, we observed a predominance of short texts, with those exceeding 512 characters making up a quarter of the total. This insight highlighted the MPU framework's suitability for subtask A. The MPU model, previously tested only on the HC3 (Guo et al., 2023) Chinese and English datasets, needed assessment for its effectiveness on multilingual datasets. To address this, we integrated the MPU framework with the XLM-Roberta (XLM-R) model to enhance its adaptability for multilingual tasks and employed stacking ensemble techniques, yielding significant improvements in our experimental outcomes.

2.2 Fine-tuning

Fine-tuning PLMs such as BERT or RoBERTa have been extensively discussed in text classification tasks (Jiang, 2023; Jiang et al., 2023, 2020). Recently, the DeBERTa model (He et al., 2020) is an enhancement built upon the foundations of BERT and RoBERTa through the incorporation of a disentangled attention mechanism and an enhanced masked decoder. We utilized the DeBERTa model and performed fine-tuning on it in our experiment. Although fine-tuning PLMs to specific domains or downstream tasks is a crucial and common practice, fully fine-tuning its large number of parameters becomes time-consuming and costly.

2.3 Adapter

In our experiments, due to the substantial size of the DeBERTa model and the size of the official datasets, each fine-tuning run required a significant amount of time. Adapter-based fine-tuning is an approach to fine-tuning a PLM that involves freezing the most of layers and inserting low-dimensional adapter modules into each layer to improve parameter efficiency. Research has shown that introducing adapters reduces the number of trainable parameters to 3.6%, with only a marginal performance drop of 0.4% (Houlsby et al., 2019). Furthermore, in some cases, models applying adapters perform even better (Bapna et al., 2019).

In our task, we employed the LoRA (Low-Rank Adapter) method (Hu et al., 2021), which injects trainable rank-decomposition matrices into each layer of the Transformer architecture, effectively freezing the PLM's weights. This significantly reduces the number of trainable parameters for downstream tasks. Therefore, we added a sequence classification head on top of the model to adapt the PLMs to the classification task. This has reduced the training costs and shortened the training time.

2.4 Stacking

Ensemble learning combines multiple base learners to form a predictive model with enhanced generalization capabilities (Sagi and Rokach, 2018). Initially, predictions are generated employing various machine learning algorithms. Then, these predictions serve as inputs for a subsequent classifier. Upon training the subsequent classifier, the integrated model is optimized to produce a new prediction set.

3 Methodology

Upon obtaining the dataset, we conducted a comprehensive statistical analysis of its scale and distribution. We observed that the multilingual training dataset contains both monolingual data and Subtask B-related data, along with an additional portion. To enhance the model's generalization capability, we opted to augment the training data through the following steps 2:

Firstly, we employed Google Translate to uniformly translate the multilingual training dataset into English. Subsequently, considering the presence of monolingual datasets within the multilingual training data and to prevent leakage of validation data, we excluded the validation dataset for monolingual tasks and Subtask B from the multilingual training dataset.

In accordance with the multi-class nature of Subtask B, we balanced the categories of the translated multilingual training dataset. We reorganized the dataset into multiple categories to better align with

the data distribution of Subtask B. To ensure data quality, we conducted thorough cleaning of the datasets.

Following data preprocessing, we separately fed the cleaned training datasets into the respective models for training. In order to leverage the strengths of different models and enhance classification accuracy, we contemplated performing stacking. We selected the top two models based on their performance on the validation dataset and saved their generated logits. Finally, we stacked these logits to obtain the ultimate results.

4 Experimental setup

4.1 Data preprocessing

The subtasks A and B involving diverse domains and sources with both human and machine-generated texts, we encountered chaotic symbols and extraneous content such as hyperlinks, numerals, and escape characters. To improve data quality, we undertook preprocessing steps including: removing special characters; eliminating excessive whitespace and line breaks; discarding Unicode escape characters and numerically formatted texts; removing hyperlinks; excluding irrelevant text lines like those for sharing, surveys, comments, ads, terms of use, and copyright notices; and deleting duplicate sentences. Notably, we avoided removing escape characters from multilingual training and validation sets to preserve original characters in non-English texts.

4.2 Data augmentation

We evaluated our models on the original dataset (**v1**) before the test set was released. We found that the multilingual set contained training and validation data for the monolingual and subtask B. To enlarge the monolingual dataset and improve model performance, we removed 5000 monolingual validation entries from the multilingual set, translated Chinese, Indonesian, Urdu, and Bulgarian data to English using Google API, and then cleaned the data to produce a refined dataset (**v2**).

we calculated the statistics of different versions of datasets, as shown in Table 1. For the multi-class subtask B, we re-labeled the dataset based on multilingual tags, addressing a severe imbalance by reducing instances in overrepresented categories for balance. After receiving the test set, we included the multilingual validation set into our training dataset and performed the same enhancement processes, creating a **v3** version for final model training and prediction.

We further analyzed the augmented version of the v2 dataset, as shown in Table 2. This analysis includes the average sentence length and average text length, as well as the proportion of texts exceeding 512 characters in length. The average sentence count was obtained through sentence tokenization using the sent-tokenize tool, while the average sentence length was calculated using the word-tokenize tool from the NLTK python library. Model performance for subtasks A and B was evaluated based on accuracy.

4.3 Monolingual models

In subtask A, submissions were made using two systems based on the different language tracks. For the monolingual English track, the system consisted of five approaches across ten models, as detailed in Table 3. The final submission system employed a stacking ensemble method, which was composed of the two best-performing models out of the ten.

In Table 3, these models had a learning rate of 1e-4, were trained for 3 epochs, and the best-performing models on the validation set were saved. For the fine-tuned models, we adhered to the inherent 512-token length limitation to ensure consistency in the input data and effective processing by the models.

The final monolingual model selected the top-performing two models, DeBERTa-v3-large and RoBERTa-base model based on MPU framework(RoBERTa-base-MPU), for stacking ensemble learning. The learning rate for the ensemble model remained set at 1e-4, trained for 1000 epochs. Only the best-performing stacking model was retained, and the final predictions were based on this optimal stacking model.

4.4 Multilingual models

In the multilingual track, we conducted experiments employing the top five models that exhibited promising performance in monolingual contexts. We opted to substitute the RoBERTa model with the XLM-R model, which is specifically designed for multilingual tasks.

Derived from the RoBERTa architecture, the XLM-R model has undergone training across 100 distinct languages, endowing it with multilingual capabilities. This versatility enables the model to process and comprehend various languages effec-

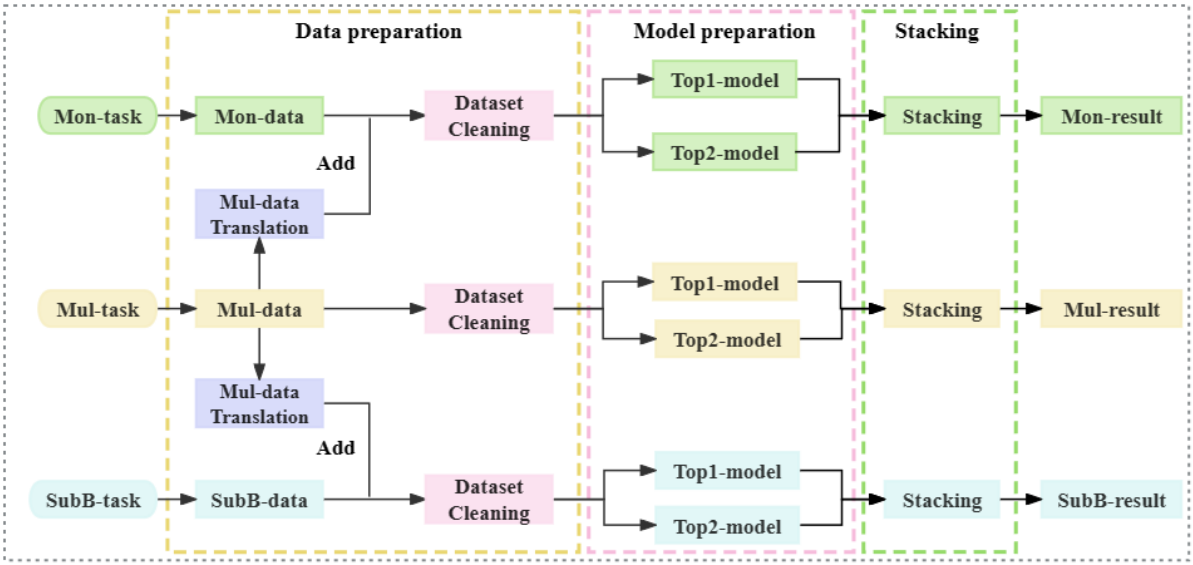


Figure 2: This is the workflow diagram for our paper."Mon-task" and "Mul-task" respectively refer to the monolingual and multilingual tasks for Subtask A, while "SubB-task" refers to Subtask B. "Mon-data", "Mul-data", and "SubB-data" respectively refer to the datasets for the corresponding tasks."Top1-model" and "Top2-model" respectively denote the models with the highest and second-highest performance among those utilized for the subtask.

Subtask	**v1**		**v2**		**v3**		
	train	dev	train	dev	train	dev	test
Monolingual	119,757	5,000	167,252	5,000	176,252	5,000	34,272
Multilingual	172,417	4,000	172,417	4,000	176,417	4,000	42,378
subtask B	71,027	3,000	105,908	3,000	176,252	3,000	18,000

Table 1: The overall data statistic. "v1" and "v2" respectively refer to the original dataset, and the dataset processed after data augmentation. "v3" refers to the training dataset that has undergone data augmentation and other processing after the official test dataset was released.

tively, leading to notable enhancements in performance across diverse cross-lingual transfer tasks. Subsequently, we identified the top two models for integration through stacking ensemble. The integration of predictions from these two models yielded superior predictive performance.

Following this, we trained the multilingual models on the v2 version of the multilingual training dataset. We set the learning rate to 1e-4 and 3 epochs while retaining the models that performed best on the validation set.

During the experimentation, we observed that the addition of adapters to the DeBERTa-v3-large model resulted in unstable performance, while direct fine-tuning of the DeBERTa-v3-large model exhibited better results. Based on this observation, we ultimately chose to integrate the XLM-R-MPU model and the DeBERTa-v3-large model. To achieve this, we saved the best models from each training session and utilized these two optimal models to generate logits. Subsequently, we merged the logits from both sets of models as part of the training data for the stacking ensemble's input linear layer. We set the learning rate to 1e-4 and extended the training epochs to 1000 to ensure thorough model training. Throughout this process, we continuously monitored and retained the best-performing stacking model, which was ultimately applied to the test set for final predictions.

4.5 Subtask B models

We extended the binary classification capabilities of the RoBERTa-base model combined with the MPU method to address multi-class problems in subtask A, employing a one-vs-rest strategy for six categories including human, ChatGPT, etc. This resulted in six separate classifiers, with classification based on the highest confidence level among positive predictions from these classifiers for each category.

Statistic	SubA-mono			SubA-multi			SubB		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
avg-sent	16.5	13.0	18.4	15.7	8.9	17.1	15.3	10.2	17.5
avg-sent-len	24.7	26.8	23.7	25.5	22.9	23.1	23.5	24.1	23.6
sent-len>512	16.7%	13.1%	23.5%	17.3%	1.2%	14.6%	15.2%	6.6%	18.9%

Table 2: The statistics of the v2 dataset. "avg-sent" represents the average number of sentences per document, "avg-sent-len" represents the average number of words per document, "sent-len>512" represents the percentage of documents that their sent-len are greater than 512 words.

Following this, we opted for the consistently excellent performance of the LoRA adapter-based DeBERTa-v3-large (DeBERTa-v3-Large-LoRA) model and applied it to subtask B. Additionally, we introduced a new adapter-based RoBERTa-large model. The model configurations were consistent with the monolingual models. In the final ensemble model, we employed stacking with the DeBERTa-v3-Large-LoRA and RoBERTa-large models. The learning rate was set to 1e-4, with a training period of 3000 epochs, and only the model with the highest score was retained.

5 Results

5.1 Monolingual results

Our evaluation work is divided into two main stages: first, the evaluation based on the officially provided monolingual dataset (**Mon1**), and second, the evaluation based on our back-translated and processed monolingual dataset (**Mon2**).

In the **Mon1** evaluation stage, we aimed for a quick baseline model implementation, using traditional deep-learning methods. The results in Table 3 on the Mon1 dataset show that traditional models generally outperformed the fine-tuned deep learning models, likely due to the small size of the official monolingual dataset, which contains only about 110,000 entries. Consequently, large models like RoBERTa-base may not be adequately trained, while smaller models such as CNN or RNN could perform better by being less prone to overfitting.

By integrating the MPU framework with the RoBERTa-base model, performance improved by 20 percentage points over direct fine-tuning, highlighting MPU's benefits in boosting short-text performance and enhancing machine-generated long text detection. Despite being designed for long documents, Longformer-base-4096 underperformed compared to CNN and Self-Attention methods on the Mon1 dataset.

The DeBERTa model, an advancement over BERT and RoBERTa, excelled in our tests, especially after fine-tuning with adapters, which improved both efficiency and performance, slightly surpassing the fully fine-tuned DeBERTa. Stacking and re-predicting logits from the top two models led to a nearly 7% improvement over the best single model, underscoring the effectiveness of model fusion in increasing prediction accuracy and stability.

In the **Mon2** evaluation stage, we retrained them on the Mon2 dataset after selecting the top five best-performing models on the Mon1 dataset. After retraining, the performance of the models on the Mon2 dataset improved by approximately 20%. likely due to shorter texts enhancing feature detection, noise reduction from removing poor-quality data, and increased dataset diversity and size. These factors combined allowed the models to gain a deeper understanding of language characteristics.

5.2 Multilingual results

Experiments performed on a refined multilingual dataset employing the DeBERTa-v3-Large-LoRA model produced a performance score of merely 0.669, notably inferior to the baseline model's performance on the unprocessed dataset. This discrepancy may stem from improperly removing crucial features during the dataset cleaning process or introducing errors. Therefore, we opted to directly train the selected model on the raw official multilingual dataset, as detailed in Table 3 above. We found that the XLM-R model integrated with the MPU framework outperformed the baseline XLM-R model by 7% on the dataset, thus confirming the effectiveness of the MPU framework.

While the BERT model excels in monolingual tasks, its performance lags behind the baseline by approximately 7% in multilingual tasks, suggesting that BERT may be less suitable for multilingual classification tasks. The DeBERTa-v3-large model, which is an improvement based on BERT

Methods	Models	Mon1	Mon2	Mul	SubB
Deep learning	CNN (Jiang et al., 2019)	0.762	-	-	-
	RNN (Lin et al., 2017)	0.729	-	-	-
	RCNN (Lin et al., 2017)	0.702	-	-	-
	Self-Attention (Jiang and Wang, 2023)	0.762	-	-	-
MPU	RoBERTa-base-MPU (**?**)	0.894	0.979	-	-
	XLM-R-MPU (Ours)	-	-	**0.798**	0.7
Fine-tuning	DeBERTa-v3-base (He et al., 2021)	0.823	-	-	-
	DeBERTa-v3-large (He et al., 2021)	0.84	0.979	0.763	-
	longformer-base-4096 (Beltagy et al., 2020)	0.737	-	-	-
	BERT (Devlin et al., 2018)	0.769	0.955	0.654	-
	RoBERTa-base (Liu et al., 2019)	-	-	-	0.75
	XLM-R (Liu et al., 2019)	-	-	0.72	-
Adapter	DeBERTa-v3-Large-LoRA (Ours)	0.843	0.948	0.669	0.858
	Roberta-large-LoRA (Ours)	-	-	-	0.862
Stacking	RoBERTa-base-MPU+DeBERTa-v3-large (Ours)	**0.96**	**0.99**	0.795	-
	RoBERTa-large+DeBERTa-v3-large (Ours)	-	-	-	**0.94**

Table 3: The overall accuracy comparison in subtask A and B. "Mon1", "Mul", and "SubB" respectively represent the accuracy of monolingual models, multilingual models, and subtask B models trained on the v1 dev set. "Mon2" is the dev accuracy on the v2 dataset.

and RoBERTa, outperforms the baseline XLM-R by 3.55% on multilingual datasets. This improvement can be attributed to DeBERTa's optimizations to both architectures, which prove particularly effective in multilingual processing, enhancing the model's learning capabilities and generalization.

In our experimental results table, we observe that the performance of the DeBERTa-v3-Large-LoRA model is 5.1% lower than the baseline model, while also exhibiting a 9.4% decrease compared to directly fine-tuning the DeBERTa-v3-large model. This discrepancy in performance may stem from significant differences in data distribution between the pre-training task and incremental training.

Specifically, there exists a substantial difference in data distribution between the DeBERTa model and the model fine-tuned via LoRA adapters, resulting in insufficient parameter updates to effectively capture these differences. This phenomenon suggests that although LoRA adapters offer a parameter-efficient fine-tuning method, relying solely on limited parameter adjustments may not suffice to achieve optimal performance in situations with substantial disparities in data distribution.

The stacking results in the multilingual task failed to surpass the performance of the XLM-R-MPU model, which could be attributed to the already robust nature of XLM-R-MPU, potentially causing the ensemble model to overfit on the training data, thereby reducing performance on validation or test data. Another possibility is that the two top-performing models exhibit high correlation in predictions (i.e., commonly making the same type of errors), thus stacking them may not yield significant performance improvements.

5.3 Subtask B results

Table 3 indicates that the performance of the XLM-R-MPU model continues to deteriorate on the dataset for subtask B, indicating poor results. This could be attributed to the model originally being designed for binary classification tasks and not being well-suited for multi-class tasks.

We found that by directly freezing the model and fine-tuning the adapter-based DeBERTa-v3-Large (DeBERTa-v3-Large-LoRA) and RoBERTa-large (Roberta-large-LoRA) models, the classification effectiveness significantly improved, outperforming the official baseline by approximately 10 percentage points.The use of the LoRA adapter allows models to more effectively utilize pre-trained knowledge while avoiding over-fine-tuning and reducing the risk of overfitting on specific tasks. After data cleaning, the pre-trained data used by DeBERTa-v3-large and RoBERTa-large were closer to the target multi-class task, potentially further enhancing their performance.

6 Ablations

We conducted ablation experiments using the DeBERTa-v3-large and Roberta-base models on the Mon1 dataset. As shown in Table 4, the experimental results indicate that the introduction of the MPU framework and stacking ensemble method significantly improves the model's performance, resulting in a notable performance enhancement

However, despite the inclusion of the LoRA adapter, the performance improvement is not significant. This could be attributed to the insufficient number of parameters fine-tuned solely by the adapter when faced with complex tasks, which hinders the model from learning additional knowledge effectively.

Methods	Results
DeBERTa-v3-large	0.84
DeBERTa-v3-large w/ LoRA	0.843
Roberta-base	0.694
Roberta-base w/ MPU	0.894
Stacking	**0.96**

Table 4: Ablation experiments. "Stacking" refers to the aggregation of results from the "DeBERTa-v3-large with LoRA" model and the "Roberta-base with MPU" model.

7 Official test results

Our system ranks 8th on Semeval 2024 Task 8 official multilingual test set of subtask A. It is noteworthy that, employing the same method, Only the model trained on the multilingual dataset surpassed the baseline of 0.80 with a score of 0.90, while models on monolingual datasets and Task B exhibited comparatively inferior performance, with none of the submissions reaching the respective domain baselines.

This phenomenon could be attributed to several factors. Firstly, although monolingual validation sets were removed from the multilingual training set, the processed multilingual training data may still share similarities with monolingual validation sets. This could lead to superior model performance during evaluation; however, due to disparities between the final test set data and the processed multilingual data, models may fail to meet baseline performance on the test set. Secondly, model training based on fine-tuning may encounter instability and catastrophic forgetting issues, thus affecting the model's generalization ability. Therefore, even with the same model, discrepancies in performance on official test sets may arise due to differences between datasets, resulting in significant performance gaps.

8 Conclusion

In conclusion, our team developed three distinct systems for SemEval-2024 Task 8, targeting the monolingual and multilingual aspects of subtask A and addressing subtask B. We achieve 8th place in the multilingual setting of subtask A. We leveraged back-translation to expand the training datasets for both monolingual and subtask B. The RoBERTa-base and XLM-R models, enhanced by the MPU framework, showed improved detection of short texts in both monolingual and multilingual settings. Finally, the stacking method allowed us to combine the strengths of multiple models, improving our system's predictive accuracy. Future efforts will consider incorporating advancements in cross-lingual pre-trained models in our subsequent work to further enhance the model's understanding of texts across diverse languages and cultural backgrounds.

Acknowledgements

This work is funded by the Natural Science Foundation of Shandong Province under grant ZR2023QF151.

References

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint.*

Zhen Guo and Shangdi Yu. 2023. Authentigpt: Detecting machine-generated text via black-box language models denoising. *arXiv preprint arXiv:2311.07700*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ye Jiang. 2023. Team qust at semeval-2023 task 3: A comprehensive study of monolingual and multilingual approaches for detecting online news genre, framing and persuasion techniques. *arXiv preprint arXiv:2304.04190*.

Ye Jiang, Johann Petrak, Xingyi Song, Kalina Bontcheva, and Diana Maynard. 2019. Team bertha von suttner at semeval-2019 task 4: Hyperpartisan news detection using elmo sentence representation convolutional network. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 840–844.

Ye Jiang and Yimin Wang. 2023. Topic-aware hierarchical multi-attention network for text classification. *International Journal of Machine Learning and Cybernetics*, 14(5):1863–1875.

Ye Jiang, Yimin Wang, Xingyi Song, and Diana Maynard. 2020. Comparing topic-aware neural networks for bias detection of news. In *ECAI 2020*, pages 2054–2061. IOS Press.

Ye Jiang, Xiaomin Yu, Yimin Wang, Xiaoman Xu, Xingyi Song, and Diana Maynard. 2023. Similarity-aware multimodal prompt learning for fake news detection. *arXiv preprint arXiv:2304.04187*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, et al. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Akim Tsvigun, Jinyan Su, Artem Shelmanov, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. MG-Bench: Evaluation benchmark for black-box machine-generated text detection.

YNU-HPCC at SemEval-2024 Task 9: Using Pre-trained Language Models with LoRA for Multiple-choice Answering Tasks

Jie Wang, Jin Wang, and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
wangjie_qpqj@stu.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

This study describes the model built in Task 9: brainteaser in the SemEval-2024 competition, which is a multiple-choice task. As active participants in Task 9, our system strategically employs the decoding-enhanced BERT (DeBERTa) architecture enriched with disentangled attention mechanisms. Additionally, we fine-tuned our model using low-rank adaptation (LoRA) to optimize its performance further. Moreover, we integrate focal loss into our framework to address label imbalance issues. The systematic integration of these techniques has resulted in outstanding performance metrics. Upon evaluation using the provided test dataset, our system showcases commendable results, with a remarkable accuracy score of 0.9 for subtask 1, positioning us fifth among all participants. Similarly, for subtask 2, our system exhibits a substantial accuracy rate of 0.781, securing a commendable seventh-place ranking. The code for this paper is published at: https://github.com/123yunnandaxue/Semveal-2024_task9.

1 Introduction

The human reasoning process includes two types of thinking: vertical and horizontal. Vertical thinking is a sequential analysis process based on rationality, logic, and rules. Horizontal thinking is a divergent and creative process. The success of language models has inspired the natural language model (NLP) community to focus on tasks that require implicit and complex reasoning. Although this type of vertical thinking task is widespread, horizontal thinking puzzles have received little attention (Jiang et al., 2024). Task 9 in the SemEval-2024 competition: brainteaser is a multiple-choice task that tests the model's ability to demonstrate horizontal thinking and challenge default common sense associations. The task consists of two subtasks, sentence and word puzzles (Jiang et al., 2023).

- Subtask 1: Sentence-type brain teaser where the puzzle defying commonsense is centered on sentence snippets.
- Subtask 2: Word-type brain teaser where the answer violates the default meaning of the word and focuses on the letter composition of the target question.

In recent years, machine learning models have garnered significant attention. Traditionally, these models have employed a two-step process involving the extraction of hand-crafted features from documents followed by classification using algorithms like Naïve Bayes (Zhang, 2004), SVM (Cortes and Vapnik, 1995), HMM (Trabelsi et al., 2012), or random forests (Ren et al., 2015). However, this approach presents limitations, such as the need for meticulous feature engineering and reliance on domain knowledge for feature design. To address these shortcomings, neural approaches have emerged. Early attempts, such as latent semantic analysis (LSA) (Dumais et al., 1988) and neural language models, initially underperformed compared to classical models but paved the way for developing more powerful embedding models. Significant advancements were made with the introduction of word2vec (Mikolov et al., 2013), ELMo (Peters et al., 1802), RoBERTa (Liu et al., 2019), GPT (Radford et al., 2019), BERT (Devlin et al., 2018), and subsequent models like GPT-3 (Brown et al., 2020) and GShard (Lepikhin et al., 2020), which boast increasingly more significant parameters and training datasets (Minaee et al., 2021).

This paper proposes a deep learning system for Task 9 in SemEval-2024, titled brainteaser. We use the decoding-enhanced BERT (DeBERTa) (He et al., 2020) model with disentangled attention as the base model and use LoRA (Hu et al., 2021) to fine-tune the model. Focal loss was used to address the issue of label imbalance. The back-translation

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 471–476

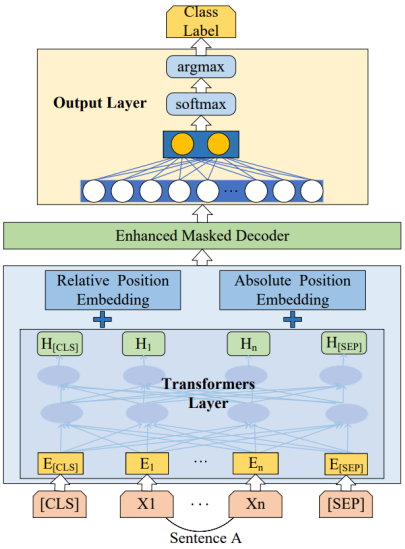


Figure 1: The overall architecture of the proposed method.

method is used to enhance the original dataset, and the processed dataset is used to train the model. The experimental results of this paper were ultimately presented in Task 9 of the SemEval-2024 competition. On the original dataset, the accuracy of Task 1 was 0.9, ranking fifth; The accuracy of Subtask 2 was 0.78, ranking seventh. The rest of this paper is organized as follows. In Section 2, we provided a detailed description of the proposed system and model. The experiment and results are discussed in Section 3. Finally, Section 4 presents the conclusion.

2 DeBERTa

Transformer (Vaswani et al., 2017) has become the most effective neural network architecture for neural language models. Unlike recurrent neural networks (RNNs) (Zaremba et al., 2014) that process text sequentially, transformers apply self-attention functionality to parallelly calculate the attention weight of each word in the input text. Therefore, compared to RNNs, they can perform large-scale model training in parallel. In this paper, the DeBERTa model we use is a new transformer neural language model that improves the Bert model using two novel techniques: a disentangled attention mechanism and an enhanced mask decoder. Figure 1 shows the structure of the system.

2.1 Tokenizer

Given a training data $\mathcal{D} = \{X^{(m)}, y^{(m)}\}_{m=1}^{M}$, $X^{(m)}$ is the input text, $y^{(m)}$ is the corresponding ground-true label, tokenizer is applied to transform $X^{(m)}$ as,

$$X = \{[\text{CLS}], x_1, x_2, ..., x_n, [\text{SEP}]\} \quad (1)$$

where x_i is the token in the text, [CLS] represents the classified characters, and [SEP] represents the terminating characters.

2.2 Encoder

DeBERTa's encoder is mainly composed of multilayer transformer encoders, and each transformer encoder is composed of multiple sub-layers. The following are the main components of the encoder of the DeBERTa model.

Token embeddings. Each token in the input text is first converted into the corresponding word embedding vector. First, we use an embedding layer to map each token x_i to its corresponding word embedding vector. The embedding matrix E is with dimension $V \times d$, where V is the size of the vocabulary and d is the dimension of the word embedding. Then, the embedding vector corresponding to the i-th token x_i can be expressed as $e_i = E[x_i]$.

Positional encoding. Positional encoding represents the absolute position of each word in the input sequence. Suppose we have a position encoding matrix P with dimensions $N \times d$, where d is the dimension of the word embedding. Then, the position encoding vector corresponding to the i-th position p_i can be expressed as $p_i = P[i]$. After adding positional encoding, the new word embedding sequence we get is $E(X) + [p_1, p_2, ..., p_N]$.

Relative positional encoding. The relative position encoding matrix is a learnable parameter matrix with dimensions$L \times 2D$, where L is the maximum sequence length and D is the word embedding dimension. In DeBERTa, the calculation process of relative position encoding is as follows:

For each pair of words (i, j), we calculate its relative position relationship vector r_{ij}.

$$r_{ij} = PE_{(i-j)} \quad (2)$$

where $PE_{(i-j)}$ represents the encoding vector at position $(i - j)$ in the relative position encoding matrix. Finally, the input text sequence X that needs to be sent to transformer encoder layers can be obtained by adding the word embedding vector, position encoding, and relative position encoding.

$$x_i = E[x_i] + p_i + r_{ij} \quad (3)$$

$$X = \{x_1, x_2, ..., x_n\} \tag{4}$$

Transformer encoder layers. Each transformer encoder layer contains the following sub-layers:

1. Multi-head self-attention. This sub-layer allows the model to focus on different parts of the input sequence simultaneously to capture global information.

2. Feed-forward neural network. This sub-layer contains a feed-forward neural network for non-linear transformation and feature extraction of the context vector at each position.

When the input text sequence passes through the encoder of the DeBERTa model, it can be expressed as:

$$H = \text{Encoder}(X) \tag{5}$$

where H is the encoded context representation, X is the input text sequence, and Encoder() is the Encoder part of the DeBERTa model.

2.3 Output Layer

In the DeBERTa model, the output layer is usually used to predict downstream tasks, such as text classification, named entity recognition, etc.

Linear transformation. First, map the output of the transformer encoder to the output space, usually through a linear transformation (fully connected layer). Assuming we have a weight matrix W and a bias vector b, the calculation of the linear transformation can be expressed as:

$$Z = H \cdot W + b \tag{6}$$

where Z is the output after linear transformation.

Activation function. The softmax function is a commonly used activation function in the output layer in multi-classification problems. The formulation of the softmax function is as follows:

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{7}$$

where the x_i is the i-th element in the input vector Z, and N is the length of the input vector. Finally, the category label is obtained by the following formula.

$$\hat{y} = argmax(softmax(H \cdot W + b)) \tag{8}$$

Focal loss. Focal loss is a dynamically scaled cross-entropy loss. A dynamic scaling factor can dynamically reduce the weight of easily distinguishable samples during training, thereby quickly focusing the center of gravity on those difficult-to-distinguish samples. The formula for focal loss is as follows.

$$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma} \log(p_t) \tag{9}$$

where the α_t is a trainable parameter, the γ is a hyperparameter, and the p_t represents the probability of the category of t obtained by softmax function.

2.4 LoRA

The low-rank adapter (LoRA) significantly reduces the number of trainable parameters for downstream tasks by freezing the weights of pre-trained models and injecting trainable rank decomposition matrices into each layer of the transformer architecture. Research has shown that the model quality and fine-tuning of LoRA on RoBERTa, DeBERTa, GPT-2 (Radford et al., 2019), and GPT-3 (Brown et al., 2020) are equivalent or better.

LoRA injects trainable low-rank matrices into transformer layers to approximate the parameter update. For a pre-trained weight matrix $W \in \mathcal{R}^{n \times d}$, LoRA decomposes the update with a low-rank factorization,

$$W + \Delta W = W + W^{down} W^{up} \tag{10}$$

where W^{down} and W^{up} are both trainable parameters. Specifically, LoRA applied such an update to the query and value projection matrix in the multi-head attention. For a specific input H_{l-1} to the linear projection in multi-head attention, LoRA can be defined as,

$$H_l = H_l + \gamma \cdot H_{l-1} W^{down} W^{up} \tag{11}$$

where γ was used to scale the contribution of LoRA.

3 Experimental Results

Datasets. The training set for subtask 1 (507 data) and subtask 2 (396 data) are processed using back-translation to enhance the model's efficiency. The dataset is translated into Chinese, Russian, Arabic, French, German, Spanish, Portuguese, Italian,

Method	Loss	Subtask 1	Subtask 2
LoRA	CE	0.93	0.51
	Focal	**0.83**	**0.67**
AdaLoRA	CE	0.37	0.37
	Focal	0.51	0.32
Prompt-Tuning	CE	0.17	0.22
	Focal	0.17	0.27
R-Drop	CE	**0.96**	**0.80**
	Focal	0.34	0.40

Table 1: Accuracy of each strategy in dev data. Results in bold are the best performance.

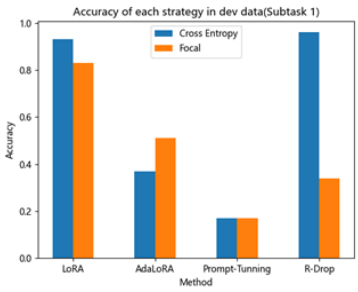


Figure 2: Accuracy of each strategy in dev data (Subtask 1).

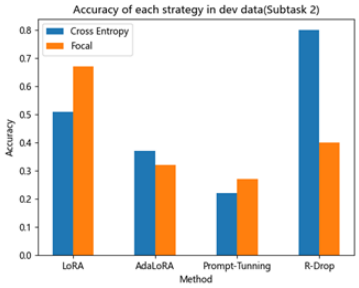


Figure 3: Accuracy of each strategy in dev data (Subtask 2).

Dataset	Subtask 1	Subtask 2
Back-translation method dataset	1.0	1.0
Original dataset	0.96	0.98

Table 2: Accuracy of LoRA in dev data.

and Japanese and re-translated into English. The translated results are added to the dataset to obtain the enhanced dataset: the training set for subtask 1 (5070 data) and the training set for subtask 2 (3960 data).

Evaluation Metrics. In this paper, the task will be evaluated based on the following two accuracy indicators.

- Example-based accuracy: treat each problem (primitive/adversarial) as a separate instance.
- Based on group accuracy: each problem and its related adversarial instances form a group.

Implementation Details. In this paper, we use the back-translation method for data augmentation to improve the model's efficiency. After obtaining more data, use focal loss to address the issue of category imbalance in the data. LoRA is used to reduce the trainable parameters of downstream tasks to save computational costs. This is the DeBERTa-V2-xxlarge model with 48 layers and a 1536 hidden size. The total parameters are 1.5B, and it is trained with 160GB of raw data.

Comparative Results. In addition to using LoRA, this paper also attempted methods such as AdaLoRA (Zhang et al., 2023), Prompt-Tuning (Lester et al., 2021), R-Drop (Wu et al., 2021), using cross-entropy and focal loss as losses, with accuracy as the evaluation metric. The results are shown in Table 1.

Figure 2 depicts validation set accuracy for subtask 1 across various methods. Models employing LoRA and R-Drop exhibit higher accuracy with cross-entropy loss. Transitioning to focal loss saw a 0.1 drop for LoRA, whereas R-Drop experienced a significant decrease. AdaLoRA's accuracy increased by 0.14 with focal loss adoption, though performance remains subpar. Prompt-Tuning's accuracy remains stagnant regardless of the loss function, indicating poor performance.

Figure 3 shows the accuracy of various methods on the validation set for subtask 2. From the figure, we can see that the accuracy of the model using LoRA increased by 0.16 after using focal loss. After using focal loss, AdaLoRA's accuracy dropped by 0.05. Moreover, no matter which method the model uses, its performance on subtask 2 is worse than on subtask 1.

Finally, we found that when using cross-entropy, R-Drop achieved the best results, with LoRA ranking second. However, after using focal loss, the accuracy of R-Drop decreased significantly. Based on the results of cross-entropy and focal loss, using LoRA yields the best result. Therefore, LoRA was chosen for model fine-tuning, and then the data augmentation dataset was used to train the model. The obtained model was retrained using the orig-

Dataset	Subtask 1	Subtask 2
Original dataset	0.900 (5)	0.781 (7)
Semantic reconstruction	0.825 (8)	0.719 (9)
Recontextualization	0.800 (7)	0.812 (6)
Original dataset + Semantic reconstruction	0.825 (8)	0.719 (9)
Original dataset + Recontextualization	0.725 (8)	0.625 (10)
Original dataset + Semantic reconstruction + Recontextualization	0.842 (12)	0.771 (13)

Table 3: Result in the Test

inal dataset, and the results on dev are shown in Table 2.

Table 3 shows the competition results on the Test, with rankings displayed in parentheses. To ensure that the task evaluates reasoning ability rather than memory ability, adversarial versions of the original data are constructed in two ways.

- Semantic reconstruction: rephrasing the original question without changing the correct answer and interfering factors.
- Context reconstruction: maintains the original reasoning path but changes the question and answer to describe the new contextual context.

As shown in Table 3, our model achieved good results on both subtask 1 and subtask 2 on the original dataset. In subtask 1, the accuracy reached 0.9, ranking fifth, and in subtask 2, the accuracy reached 0.781, ranking seventh. Except for the final dataset, the accuracy of our model ranks in the top ten. Moreover, our model performs better on subtask 1 except for the context reconstruction dataset. It is well proven that our system has demonstrated competitive performance.

4 Conclusions

This paper describes a deep learning model for a multiple-choice task (Task 9: brainteaser in the SemEval-2024 competition), using DeBERTa as the base model and achieving good results, ranking fifth in accuracy in subtask 1 and ranking seventh in accuracy in subtask 2. However, there is still considerable room for improvement in the model. Therefore, we will try more methods to improve the model's efficiency in the future.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20:273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Susan T Dumais, George W Furnas, Thomas K Landauer, Scott Deerwester, and Richard Harshman. 1988. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 1802. Deep contextualized word representations. corr abs/1802.05365 (2018). *arXiv preprint arXiv:1802.05365*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. 2015. Global refinement of random forest. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 723–730.

Chiraz Trabelsi, Bilel Moulahi, and Sadok Ben Yahia. 2012. Hmm-care: Hidden markov models for context-aware tag recommendation in folksonomies. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 957–961.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Harry Zhang. 2004. The optimality of naive bayes. *Aa*, 1(2):3.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

Team jelarson at SemEval 2024 Task 8: Predicting Boundary Line Between Human and Machine Generated Text

Joseph Larson
Indiana University
joelarso@iu.edu

Francis Tyers
Indiana University
ftyers@indiana.edu

Abstract

In this paper, we handle the task of building a system that, given a document written first by a human and then finished by a large-language model (LLM), the system must determine the transition word, i.e. where the machine begins to write. We built a system by examining the data for textual anomalies and combining a method of heuristic approaches with a linear regression model based on the text length of each document.

1 Introduction

Large Language Models (LLMs) have never been more available than they are today. The consequence of this is an increase in machine-generated content within various domains. While some of this content could be considered useful, concerns related to the abuse of LLMs has arisen, e.g. the generation of fake product reviews (Adelani et al., 2019), spamming/phishing schemes (Weiss, 2019) and fake news generation (Zellers et al., 2019; Brown et al., 2020; Uchendu et al., 2020). Weiss (2019) demonstrated that humans can only detect human generated text from machine generated text at chance level. This illustrates the clear need for automatic systems to detect LLM generated content.

Regarding mere impressionistic differences between the two types of text, it has been observed that LLMs tend to be more focused, i.e. never leaving the subject matter of their prompt, more objective and highly formal. Their human counterparts tend to be less formal, with more propensity to stray from the topic at hand and more emotional. In terms of linguistic differences between the two, humans use less nouns and conjugations, while employing more punctuation and adverbs. Dependency relations are shown to be shorter. Lastly, human texts have more types in texts of the same length (Guo et al., 2023).

An assumption many researchers take is that LLMs is that language models sample from the head to generate natural looking text e.g. max sampling (Gu et al., 2017) and k-max sampling (Fan et al., 2018). (Solaiman et al., 2019) use a bag-of-words approach with tf-idf feature vectors (both unigrams and bigrams) and a logistic regression model to differentiate between human-written web pages and text generated web pages from GPT2. They examine a different number of parameters of the LLM (117M, 345M, 762M and 1,542M) as well as different sampling methods (k-sampling (sampling the highest probability tokens until a threshold of specified tokens is reached), p-sampling (sampling from the smallest possible set of words until a cumulative probability is reached) and pure sampling (also known as temperature sampling, where lower 'temperatures' are associated with higher probabilities for tokens). Their findings are that the larger the LLM, the harder to detect how machine-like the generated text is and k samples are easier to detect than pure samples, probably due to the fact that k samples over-produce common words, which is easy to detect using statistical methods.

Gehrmann et al. (2019) use BERT and a group of statistical features: the probability of each word, absolute rank of each word and the entropy of the distribution and create a tool for users to see specifically what features are more likely to be machine generated over human generated. They clearly show that the model GPT-2 oversamples certain words; it is worth pointing out, however, that as LLMs grow more sophisticated, such methods might not work as well. Solaiman et al. (2019) use fine tuning on RoBERTa and finds it can detect text generated from GPT-2 with an accuracy of 95%. The most noteworthy aspect of this study is that fine-tuning on GPT-2 itself did not yield as impressive results, which contradicts Zellers et al. (2019) findings which allude to the idea that the best detector of text generated from LLMs are the LLMs themselves. The RoBERTa detector has also been used in detecting fake news articles from several LLMs (Uchendu et al., 2020), Amazon product reviews

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 477–484

(Adelani et al., 2019) and biomedical texts (Rodriguez et al., 2022).

2 Task

This task is slightly different from the tasks described in the previous sections, since the purpose of this task is to guess the correct index at which the LLM starts writing. Since it no longer a binary classification task, i.e. given a document guess if it is a human or machine who wrote it, (which should be approached as an authorship attribution task), it was deemed helpful to examine other computational tasks whose purpose is to generate a boundary line in documents. King and Abney (2013) used four different classifiers to classify words in bilingual documents. They found that Naive-Bayes worked the best using either 1-5-grams, both character and word level. Lui et al. (2014) used a similar Bayesian model to detect language segments in multilingual documents, using byte-encoded n-grams as features and achieving the best results with higher-resource languages like English. Although the task here is profoundly different from these two previous examples, we took inspiration from these studies believing there must be linguistic differences that can be detected with statistical methods between the human generated text and the machine generated text.

3 Data Examination

Anomaly	Frequency	Location
word..word	875	Transition
^␣.Word	252	Transition
single line break	2,334	Human
double spacing	599	Human
gratuitous spacing	65	Human
2× 5-gram	1558	Machine*
2× 10-gram	382	Machine*
2× 15-gram	160	Machine*
2× 20-gram	96	Machine*
3× 5-gram	115	Machine*

Table 1: Anomalies found in training data, where 2× indicates that a particular n-gram appears twice in sequence. Machine* denotes that these occurred overwhelmingly in the machine text (with exceptions being under 1%).

The dataset for this task is the same from (Wang et al., 2024). We created a script to manually examine the transition words for all documents. One striking feature of the data is that only the human generated text featured single line breaks. Another was that in many cases the transition word occurred after tokens which had a word, followed by two full stops and another word. Table 1 provides a full list of anomalies found in the training and development set with their their respective frequencies. Some of the anomalies were present only in the human written text or occurring at the transition word token. Others were found mostly in the machine generated text. The anomalies that were featured near or around the transition word resulted as the most predictable for the creation of our model.

We also examined how frequent each transition token was in the corpus and how often it occurred as a transition word. Since the final evaluation was in terms of the distance from the actual index where the transition word occurred using the formula $\text{text} = \text{document.split(' ')}$, we decided to include tokens with different case and punctuation as separate tokens. Table 7 in the appendix contains the most frequent transition words, all appearing as transition words at least 30 times in the data set, as well as their relative frequency in the training data overall.

4 Experiments

For all experiments, scores are reported as a mean of the results of three runs ± the standard deviation, assuming random choice was involved somehow.

4.1 Random Choice

To establish our own baseline, we decided to create a system that randomly chose an index between 0 and the length of the split text. We then chose various coefficients to multiply the text length by. Table 2 gives a complete list of these experiments. Dividing the length by half or around half e.g. 0.4 gave the lowest MAE, suggesting that the majority of indexes are towards the beginning halves of the texts, not towards the end.

4.2 Heuristics

Based upon the anomalies we found in the data, we decided to incorporate explicit rules in our system of random choice. The first rule that that we experimented with was having the system guess the position of the transition word as the the one proceeding any token that had the pattern Word..Word. While experimenting with the exact regular expression pattern to use, we found the one that accurately guessed the correct index every time was $\backslash\text{w}\backslash.\backslash.\backslash\text{w}$. After this, we established a similar pattern found in the dataset is that when the document's first token was a space, full stop and then a word, then the

Upper bound	MAE
1 len(t)	75.4 ± 0.342
0.5 len(t)	41.6 ± 0.247
0.33 len(t)	43.0 ± 0.287
0.25 len(t)	47.2±0.111
0.66 len(t)	48.2±0.265
0.75 len(t)	53.9±0.246
0.4 len(t)	41.5±0.661
0.6 len(t)	44.6±0.788

Table 2: Random Choice Experiments for training data. The upper bound column indicates the upper bound of the random choice from $0...n$. Margin of error is given for the mean of three trials.

document was entirely machine generated, meaning the correct index was 0. After incorporating these two rules into our system, we left them in all subsequently tested systems, since their predicative power was completely accurate. After establishing these two baseline rules, we investigated having the system guess a random position after the last single line break in the document, guessing the transition word as being one of the frequent transition words under a certain threshold of relative frequency (< 0.01, 0.005, 0.001) and guessing the position as starting with the second repeated n-gram.

Table 3 provides a summary of all heuristic experiments. In the case that a certain rule did not apply to a given document, a random position between 0 and half the length of the text was guessed. The first two rules mentioned reduced the best score from the previous experiments by over 10 MAE, demonstrating they were by far the most robust. Guessing the index as being after the last line break improved the MAE by over 2, indicating it also had a slight effect on overall accuracy.

4.3 Linear Regression

We investigated a heuristic model based upon the length of the text. We were able to combine the first two selected rules with other rules based on text length of each document. The best MAE we obtained from doing this was 25.045±0.231. We decided to investigate using a linear regression model based upon text length, since $R^2 = 0.659$. Figure 1 shows the distribution of the index positions in the training set based on text length. The first experiment combined the first two rules of the previous section and predictions based on a linear regression model for all documents that these rules did not apply to. We obtained a baseline MAE of 20.464 for this. Figure 2 shows the distribution of our guesses using this baseline. We created several different linear models, based upon text length and well as excluding non-heuristic data points and found that our baseline performed the best on the training data. We also combined some heuristic methods from the previous section with this model and found they mostly performed worst, with the exception of slightly modifying the predictions for the texts with a length over 975 (since these are mostly outliers), in which case this method performed slightly better than baseline. Initially, we trained our model on the train data and tested on the dev set provided by the organizers. We obtained slightly better this way, with our baseline model obtaining 18.9 MAE on the dev set. Figure 3 shows the distribution of the dev set by text length and index. Figure 4 shows our predictions for the dev set. They are much more linearly distributed than the overall train set, which

i	Else?	MAE
Last \r\n+1	2nd 10 gram	50.0±0.123
Last \r\n+1	2nd 15 gram	49.2±0.321
2nd 10 gram	Last \r\n+1	47.4±0.412
2nd 15 gram	Last \r\n+1	46.2±0.374
2nd 5 gram	Random Choice	39.1±0.212
$f < 0.001$	Random Choice	32.5±0.232
Word..Word	Random Choice	32.2±0.542
␣.Word	Random Choice	30.8±0.214
Last \r\n+1	$f < 0.005$	30.1 ±0.401
2nd 10 gram	Random Choice	30.1±0.341
2nd 15 gram	Random Choice	29.8±0.021
\r\n+1	Random Choice	28.2±0.439
Last \r\n+1	$f < 0.001$	28.2±0.436
Last \r\n+1	$f < 0.01$	28.2±0.303

Table 3: Heuristic Experiments. i stands for index and f stands for relative frequency i.e. the proportion a token appeared as a transition word to how often it appeared in the data overall. The i column refers to what was guessed as the index first. If this feature was not present in the document, the Else? column indicates what was guessed for that document instead. $f < n$ refers to a transition word with absolute frequency less than n that was guessed as the index. We first tested the Word..Word rule then the ␣.Word rule. We found they always yielded the correct index so we included them in all subsequent experiments. Results are still given in descending order of MAE. Beyond this, all experiments were independent, not cumulative.

i	Else?	MAE
Baseline	N/A	20.5
Non-heuristic data	N/A	28.7
len(t) $<$ 650	N/A	20.4
len(t) $<$ 1000	N/A	20.5
len(t) $<$ 800	N/A	20.4
Random Choice*	Baseline	20.4$\pm$0.009
Last \r\n	Baseline	43.3
2nd 10-gram	Baseline	21.6

Table 4: Linear Regression Experiments. The baseline model refers to a linear model for all data. In some cases, a linear model was created for only some data. The i column refers to index that was chosen first e.g. in the case of the baseline model it was always the either the two previously mentioned heuristics or the index predicted by the linear model. *Applied a random index between 600 and 700 for this experiment for len(t) $<$ 950

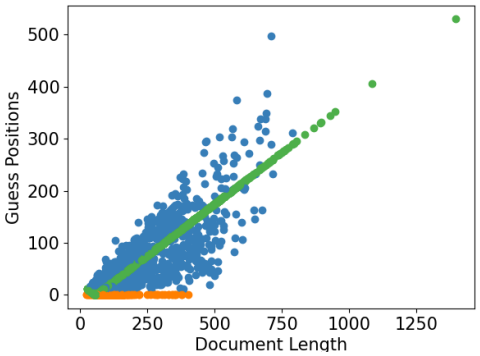


Figure 2: Distribution of Training Data by Text Length (X-Axis) and Our Predicted Index of Transition Word Using Linear Regression and Heuristics. (Y-Axis) Blue corresponds to guesses based on the rule Word..Word, orange corresponds to guesses based on the ^␣.Word rule and green corresponds to guessed made with linear regression.

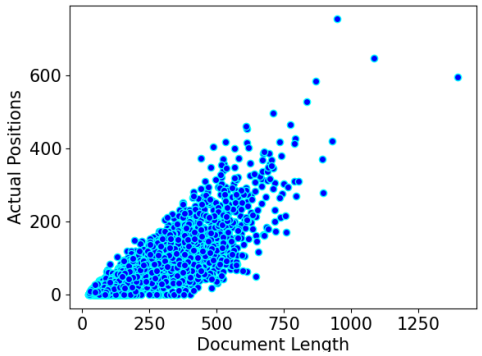


Figure 1: Distribution of the Training Data by Text Length (X-Axis) and Index of Transition Word (Y-Axis)

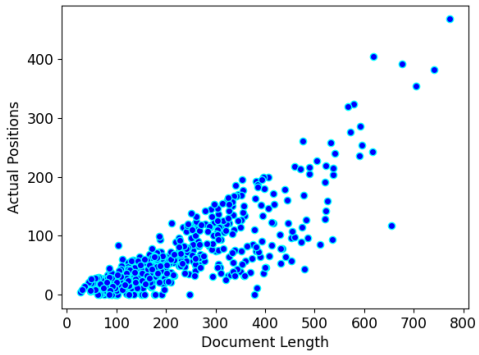


Figure 3: Distribution of the Dev Data by Text Length (X-Axis) and Index of Transition Word (Y-Axis)

explains why the model performed slightly better. Table 4 shows the results for all experiments using linear regression.

5 Results

The solution we submitted to the contest was our baseline linear regression model, since it performed the best, with the exception of the one model with the outlier rule. We chose this over the latter since we assumed the test data would have less outliers in terms of text length, so the baseline linear regression model might perform the best. Our final score for our submission was an MAE of 48.139.

We first examined the test data for the same anomalies found in the training data. Table 8 in the appendix gives a summary of these anomalies. There are far fewer transition word anomalies than

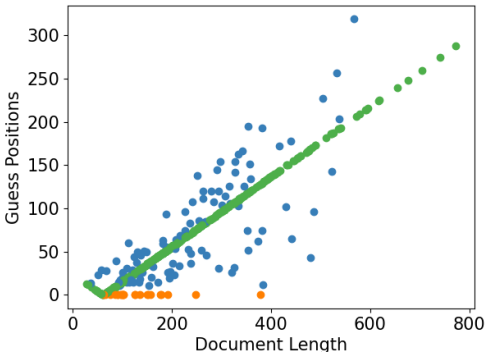


Figure 4: Distribution of Dev Data by Text Length (X-Axis) and Our Predicted Index of Transition Word Using Linear Regression and Heuristics (Y-Axis) Blue corresponds to guesses based on the rule Word..Word, orange corresponds to guesses based on the rule ^␣Word and green corresponds to guessed made with linear regression.

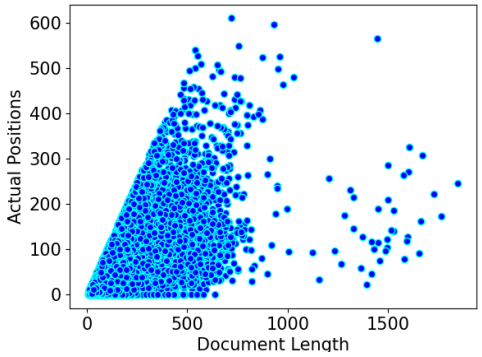


Figure 5: Distribution of Data by Text Length (X-Axis) and Actual Position (Y-Axis) in Test Data

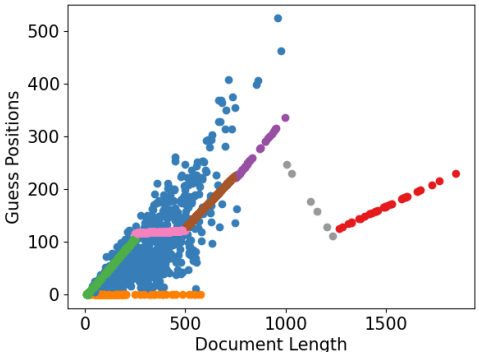


Figure 6: Distribution of Data by Text Length (X-Axis) and Predicted Position of Baseline Model (Y-Axis). Blue corresponds to guesses made with Word..Word rule, orange corresponds to guesses based on our ^␣Word rule, green corresponds to the first linear regression model, pink to the second, brown to the third, purple to the fourth, grey to the fifth and red to the sixth.

in the training data, while the number of repeat n-grams is much higher than in the training data. We also looked at the most frequent transition words in test data. Table 9 in the appendix shows a complete list of all frequent transition words that occurred more than 50 times in the training data, along with their relative frequencies.

Figure 4 shows the distribution of the indexes based on text length for test data. It is much less linearly distributed than the training data (R^2 = 0.26471) and contains a lot more outliers, which explains why our linear model performed much more poorly on it. The next sections explain experiments we did with the training data to improve the linear and heuristic model.

6 Post Hoc Data Analysis

Since the test data is less linearly distributed than the training data, we decided to try different linear models for different lengths of text. Nonetheless, we still trained a linear model on the test data to get a baseline for subsequent experiments. We obtained a baseline of 44.6 MAE. After, we tried different fitting the data to a different number of linear models based upon different text lengths. We used up to six different models in each experiment, adjusting bin sizes. Ultimately, we fit each bin to correspond to the number of quintiles for the model e.g. for final bimodal model, we used the threshold as the median. Our best result ended up being a sixmodal model with the bins 250, 500, 750, 1000 and 1250. We decided to use this linear model when applying subsequent heuristic methods. Table 5 provides the results of our linear experiments. Figure 6 shows a scatter plot of our guesses.

Regarding heuristics, we found that of the frequent transition words in the test data, the ones that almost always occurred as the transition word in a document containing it were those with a relative frequency in the corpus under 0.0001, with the exception of commas and empty strings. Combining this rule with the baseline linear model reduced MAE by ~2. We then looked at repeat higher order n-grams with worse performance. Even though the machine generally generated repeat higher order n-grams, it was still not predictive when determining the boundary line. Lastly, we looked at frequent transition bigrams and frequent transition trigrams. Setting these as the index when they occurred in a document only improved our score slightly. More than anything, they were more accurate in picking the index if they occurred at the beginning of the document, in which case the index for that document was 0. Table 6 provides a summary of all heuristic modifications we made to our system.

7 Conclusion and Considerations for Future Tasks

For our system, due to time constraints, we did not perform any state of the art techniques and of course did not obtain any state of the art results. However, what our paper demonstrates above all is the need for both training data and test data to be better processed with as few textual anomalies as possible. For those teams who trained a neural model for this task, it would be interesting to see what the model learns if these anomalies are removed from the test and training data. It is our hypothesis that

Model	Bins	MAE
Baseline	N/A	44.6
Bimodal	200	42.7
Bimodal	212	42.8
Bimodal	250	42.8
Bimodal	500	44.0
Bimodal	750	44.0
Trimodal	750, 1000	44.0
Trimodal	148, 301	42.7
Fourmodal	125, 212, 358	42.7
Fivemodal	111, 171, 261, 395,	42.6
Sixmodal	103, 149, 212, 299, 423	42.6
Sixmodal	250, 500, 750, 1000, 1250	42.5

Table 5: Linear Regression Experiments, Training and Testing on Test Data. Bins indicate cut off points for models within that particular text length.

Model	*i*	*f*	MAE
Baseline	FTW	< 0.0001	42.6
Baseline	FTW	< 0.001	42.6
Baseline	FTW	< 0.005	71.6
Baseline	FTW	< 0.002	53.0
Sixmodal	FTW	< 0.0001	41.3
Sixmodal	FTB	< 0.0001	41.2
Sixmodal	0 if FTB	< 0.0001	41.1
Sixmodal	FTT	< 0.0001	41.1
Sixmodal	0 if FTT	< 0.0001	41.0

Table 6: Heuristic Adjustments to Linear Models. i indicates index, f indicates relative frequency, FTW indicates Frequent Transition Word, FTB indicates Frequent Transition Bigram and FTT indicates Frequent Transition Trigram.

removing these anomalies would worsen the performance of neural models. Since the best models in this shared task received a score of around 16.0, it would also be interesting to see what kinds of texts they scored better on. Our hypothesis is that texts with the mentioned anomalies were easier to detect for neural networks and that they had more difficulty with longer texts, since longer texts were not featured in the training set. Not knowing the exact LLM used to generate the machine generated text for this dataset it is difficult to say with certainty, but it also our hypothesis that a more sophisticated statistical model could potentially detect more differences between the human and machine text by examining the sampling frequency of words to determine a more probable boundary line.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Jiatao Gu, Kyunghyun Cho, and Victor O.K. Li. 2017. Trainable greedy decoding for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1978, Copenhagen, Denmark. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Ben King and Steven Abney. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1110–1119, Atlanta, Georgia. Association for Computational Linguistics.

Marco Lui, Jey Han Lau, and Timothy Baldwin. 2014. Automatic detection and language identification of multilingual documents. *Transactions of the Association for Computational Linguistics*, 2:27–40.

Juan Diego Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233, Seattle, United States. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Max Weiss. 2019. Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions. *Technology Science*, 2019121801.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. *Defending against Neural Fake News*. Curran Associates Inc., Red Hook, NY, USA.

8 Appendix

Token	TW Frequency	Rel. Frequency	Token	TW Frequency	Rel. Frequency
the	375	0.07279	The	261	0.01178
paper	238	0.01176	ϵ	119	0.00405
proposed	47	0.00568	authors	131	0.00605
This	30	0.00443	in	32	0.01612
is	58	0.01310	a	44	0.02188
of	71	0.03655	this	40	0.00713
and	57	0.03038	it	43	0.00596
to	69	0.02741	␣.The	197	0.00020
However,	54	0.00167	I	44	0.00241
␣.In	34	0.00004			

Table 7: Frequency of Transition Words in Training Data. TW indicates Transition Word, i.e. how often a particular word appeared as a transition word. Relative Frequency refers to the ratio of how many times a word appeared in the training data over how many tokens in training data n = 987,374).

Anomaly	Location					Frequency
word..word	Transition					626
^␣.Word	Transition					97
single line break	N/A					3,555
double spacing	Human					981
gratuitous spacing	Human					92
n-**grams:**		$2\times$	$3\times$	$4\times$	$5\times$	$6\times$
5-gram	Machine*	4,180	1,447	609	282	167
10-gram	Machine*	1,444	270	100	65	—
15-gram	Machine*	674	93	48	42	—
20-gram	Machine*	352	46	38	30	—

Table 8: Anomalies found in test data, where $2\times$ indicates that a particular n-gram appears at least twice. Machine* denotes that these occurred overwhelmingly in the machine text (with exceptions being under 1%).

Token	Tw Frequency	Rel. Frequency	Token	Tw Frequency	Rel. Frequency
I	167	0.00433	The	257	0.00637
""""\nThe	100	<0.00001	the	485	0.05645
authors	203	0.00354	is	94	0.01476
would	74	0.00422	have	77	0.00563
.The	68	< 0.00001	However,	204	0.00139
\n\nPlease	141	< 0.00001	they	53	0.00515
this	73	0.00474	a	93	0.02312
In	77	0.00132	of	191	0.02941
there	83	0.00226	that	66	0.01293
in	65	0.01422	to	187	0.03286
be	60	0.00867	,	57	<0.00001
It	114	0.00197	For	81	0.00144
paper	186	0.00482	ϵ	270	0.00162
This	129	0.00415	\n\nThe	53	0.00021
and	87	0.02577	They	62	0.00124
\nthe	69	< 0.00001	for	64	0.00974

Table 9: Frequency of Transition Words in Test Data. TW indicates Transition Word, i.e. how often a particular word appeared as a transition word. Relative Frequency refers to the ratio of how many times a word appeared in the training data over how many tokens in training data n = 2,838,565).

HU at SemEval-2024 Task 8A: Can Contrastive Learning Learn Embeddings to Detect Machine-Generated Text?

Shubhashis Roy Dipta
Department of CSEE
University of Maryland, Baltimore County
Baltimore, Maryland, USA
sroydip1@umbc.edu

Sadat Shahriar
University of Houston
Houston, Texas, USA
sshahria@cougarnet.uh.edu

Abstract

This paper describes our system developed for SemEval-2024 Task 8, "Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection" Machine-generated texts have been one of the main concerns due to the use of large language models (LLM) in fake text generation, phishing, cheating in exams, or even plagiarizing copyright materials. A lot of systems have been developed to detect machine-generated text. Nonetheless, the majority of these systems rely on the text-generating model. This limitation is impractical in real-world scenarios, as it's often impossible to know which specific model the user has used for text generation. In this work, we propose a **single** model based on contrastive learning, which uses ≈**40% of the baseline's parameters** (149M vs. 355M) but shows a comparable performance on the test dataset (***21st out of 137 participants***). Our key finding is that even without an ensemble of multiple models, a single base model can have comparable performance with the help of data augmentation and contrastive learning. [1]

1 Introduction

In recent years, Natural Language Processing (NLP) has been totally dependent on Deep Learning rather than statistical machine learning. With multi-task learning (Caruana, 1997), attention-based transformers (Vaswani et al., 2017), and the use of Reinforcement Learning in NLP (Christiano et al., 2017), it has been used in our day-to-day life from mathematical calculations (Yang et al., 2023) to email writing. But with huge help, it has also been used to generate fake news (Zellers et al., 2019), to plagiarize copyright materials (Dehouche, 2021), and also to cheat in exams or assignments (Cotton et al., 2023; Fyfe, 2023). Humans can identify machine-generated text only at the chance level (Jawahar et al., 2020). There has been a dire need to develop a system to detect machine-generated text.

Though a lot of works (Badaskar et al., 2008; Gehrmann et al., 2019; Zellers et al., 2019; Jawahar et al., 2020; Ippolito et al., 2020; Chakraborty et al., 2023; Pu et al., 2023; Mitchell et al., 2023; He et al., 2023; Guo et al., 2023) have already been deployed for detecting machine-generated text, with the current development of LLMs, most of the systems are failing to find out which one is human-generated vs. machine-generated (mostly due to the improvement of coherency, fluency and usage of real-world dataset (Radford et al., 2019)). In this context, the task "Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection" provides a dataset for training models to classify machine-generated texts. The shared task consists of three sub-tasks: Binary Classification (Machine vs. Human), Multi-class Classification (Which model/human generated this?), and Span Detection (Which part of the text is machine-generated?). A detailed description of the task can be found in the shared task paper (Wang et al., 2024).

In this paper, we describe our final submission on Subtask A (Binary Classification). There were two big challenges of this task: **First,** five Different models have been used to generate the machine-generated text. Zellers et al. (2019) has shown that the best defense for machine-generated text is the model itself that was used for generation. However, in reality, there is a massive surge in large language models (LLMs), each with its own unique style of text generation. The challenge in this particular subtask has heightened due to the utilization of five different LLMs. This complexity demands a versatile, model-agnostic architecture capable of detecting text generated by LLMs in a generalized manner. **Second,** Following the previous challenge, the organizers have employed a different model

[1] Our code is publicly available at https://github.com/dipta007/SemEval24-Task8

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 485–491

for generating the validation and test datasets compared to those used in the training set. This implies that the text was drawn from a completely distinct distribution. As a result, participants must develop a generalized model capable of performing effectively regardless of the specific model used in the text generation process.

In response to the key challenges, we have investigated the performance of contrastive learning for this particular task. Contrastive learning has been used as a valuable technique across various domains, including Text Embedding (Neelakantan et al., 2022), Document Embedding (Luo et al., 2021), Event Embedding (Roy Dipta et al., 2023), vision (Chen et al., 2020) and Language-Vision learning (Radford et al., 2021). Notably, unlike the majority of submissions in any shared task like competition, Our final submission utilized a **single** model to classify the machine-generated texts rather than an ensemble of multiple models. Hence, our contributions to this paper are as follows,

1. We proposed a novel data augmentation technique, which nearly makes the data X times bigger (X is the number of models used for data augmentation).

2. We propose a single unified model that shows a comparable performance on the test dataset.

3. We have shown that even with a single model, contrastive learning with data augmentation shows a comparable performance, which opens up a door for future exploration.

2 Related Works

In this section, we will provide the prior works that have been done in the realm of machine-generated text detection (§2.1) and contrastive learning (§2.1).

2.1 Machine Generated Text detection

With the progress of LLMs, much prior research has been done to counter-attack the misuse of the LLMs. Before the attention and transformers, Badaskar et al. (2008) has shown how the syntactic and semantic features can help in classifying between human and machine-generated text. Later, Gehrmann et al. (2019) has provided a statistical detection system based on the assumption that the machine samples from the high probability words through max sampling (Gu et al., 2017), k-max sampling (Fan et al., 2018), beam search (Shao et al., 2017). So, the authors used the probability, rank, and entropy of words as features to classify a machine-generated text. Jawahar et al. (2020) has shown that state-of-the-art LLM can generate texts with human-like fluency and coherence without grammatical or spelling errors. Lastly, Mitchell et al. (2023) have used the change of log-probability between the original text and after random perturbation.

2.2 Contrastive Learning

Contrastive learning was first introduced in the visual domain (Chen et al., 2020). Later, it has been widely used in NLP for representation learning (Xu et al., 2023; Wang and Dou, 2023), event similarity tasks (Gao et al., 2022) and event modeling (Roy Dipta et al., 2023). Inspired by the latter works, we have explored whether contrastive learning can help in machine-generated text detection.

3 System Overview

Our system is divided into three parts: where the first part is data augmentation (described on §3.1), the second part is contrastive learning (described on §3.2), and the last part is the classification head (described on §3.3) over the document embeddings.

3.1 Data Augmentation

The dataset provided in the shared task has text and their corresponding label. However, we need a positive and a (hard) negative pair to use contrastive learning. Our main inspiration for using contrastive learning is that as the texts come from two different entities (machine vs. human), the embedding space should also be different. To facilitate the task, we have used a paraphrase model to generate alternate texts for each text in the dataset. In that way, now, every instance of the dataset has one human/machine-generated text and one machine-generated text. We have utilized the human-generated text as the hard negative [2] and the machine-generated text as the soft positive [3].

Another challenge we faced during the paraphrasing of the dataset is that the texts are long. If we give the whole text to the paraphrase model and ask for alternate text, it gives a much shorter text (an issue we observed in the used paraphrase model). In our primary validation, that gives bad

[2] Hard negatives are the total opposite of the given text

[3] Soft positives expressed the same idea but might not be the exact one

results due to the loss of information while shortening the text. So, instead of giving the whole text at once, we have split the data by end-of-sentence or newline. Then, each sentence was paraphrased on its own and then joined together again to get the previous structure. The technical details behind generating paraphrases and using them for contrastive learning have been discussed in §4.1 and §4.2, respectively.

3.2 Contrastive Learning

With the availability of an appropriate dataset for contrastive learning, we proceeded to develop our model. Our main assumption was that the embedding of the machine-generated text and human-generated text would exhibit significant differences. A simple overview of the model is shown in the Fig. 1.

The positive and negative data go through the same shared encoder to generate an embedding. This embedding is then used in contrastive learning. We have used the following loss formulation for our contrastive learning:

$$\begin{aligned}\mathcal{L}_{con} &= (1-y) * cos(x_1, x_2) \\ &\quad + y * max(0, cos(x_1, x_2))\end{aligned} \tag{1}$$

Here, x_1 and x_2 are the embeddings of two different pairs, respectively. $cos(x_1, x_2)$ is the cosine-similarity score between two embeddings. y is $+1$ for positive-positive pairs and -1 otherwise. In our task, y is $+1$ if the data instance contains text from a machine and the other is paraphrased text and -1 if the given text is from a human and the other is paraphrased.

3.3 Classification Loss

In contrastive learning, our primary objective is to acquire meaningful embeddings containing sufficient information for distinguishing between human-generated and machine-generated text. However, we also need to use a classifier model for the downstream task of outputting the actual label. Keeping that in mind, we have used a simple two-linear layer classifier head on top of the embeddings generated by the encoder. During inference time, we used this classifier head to output the labels. We have optimized our model using a simple binary cross-entropy (BCE) loss.

The total loss of our model is defined as,

$$\mathcal{L} = \alpha * \mathcal{L}_{con} + \beta * \mathcal{L}_{cls_+} + \gamma * \mathcal{L}_{cls_-} \tag{2}$$

Here, $\mathcal{L}_{cls_+}$ is the BCE loss of the positive example, and $\mathcal{L}_{cls_-}$ is the BCE loss of the negative sample of the data instance. α, β, and γ are hyperparameters that were set to 0.7, 0.8, and 0.1, respectively, based on validation data.

4 Experimental Setup

The following sections are used to describe the technical details behind our data augmentation technique (§4.1), Encoder (§4.2), Classifier Head (§4.3) and Hyperparameters (§4.4).

4.1 Data Augmentation & Pre-processing

We preprocess the raw input, splitting each document into multiple sentences for paraphrasing. After the preprocessing, we got $\approx$ 3.6 million sentences. Even if we are splitting by new lines or end-of-sentences, we kept exactly the same format during joining, i.e., two new lines rather than 1, to keep most information intact. As the paraphrasing is done on the sentence level rather than the paragraph level, the number of paraphrased sentences is the same as the input sentences (3.6M). So, ideally, we got double the number of training data just by using the data augmentation.

We have tried multiple models from HuggingfaceHub [4] [5] to generate paraphrase. In our final submission, we have used Bandel et al. (2022)'s model [4] for our data augmentation. Use of multiple models or use of prompt-based models (Achiam et al., 2023; Touvron et al., 2023) for data augmentation has been left out for future exploration due to time and compute constraints. For data split, we use the official train & dev data split. Only train data is used for data augmentation, and the dev data is used to calculate evaluation metrics.

4.2 Pre-trained Encoder

To encode the document, we have used a pre-trained version of longformer-base (Beltagy et al., 2020) [6]. The reason behind using this encoder rather than others is, **One,** longformer is good for getting embeddings for long documents because of using global vs. local attention (more details in Beltagy et al. (2020)). **Second,** the pre-trained version was fine-tuned for paraphrase detection, which is kind of similar to our task.

[4] ibm/qcpg-sentences
[5] ceshine/t5-paraphrase-paws-msrp-opinosis
[6] jpwahle/longformer-base-plagiarism-detection

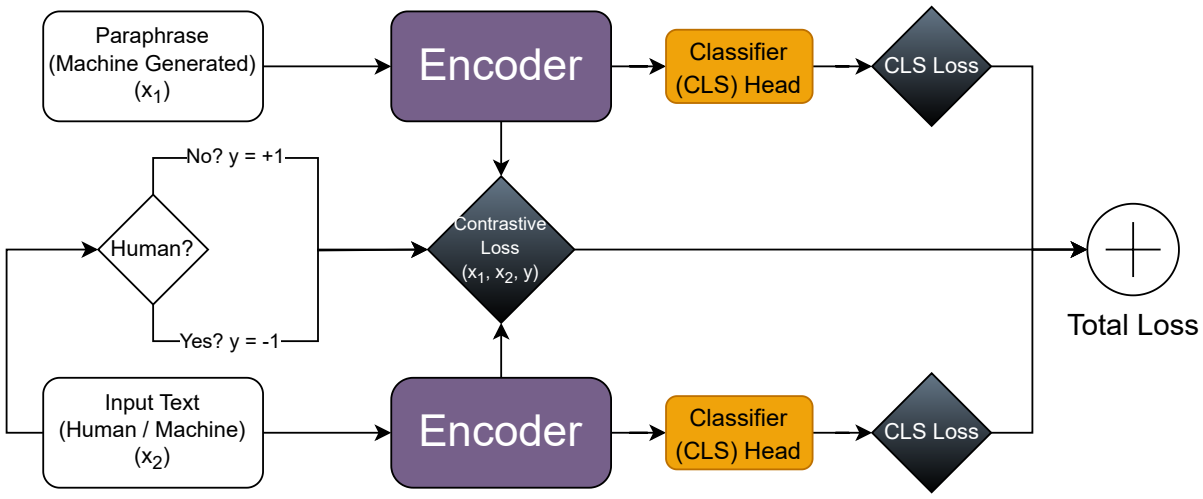


Figure 1: Overview of our model architecture. The same color weights are shared (encoder & classifier head). Diamond boxes represent the loss function, and the plus sign represents the summation of the three losses. The input to the contrastive loss depends on the original label (y=+1 if human, else −1).

4.3 Classifier Head

We have used two linear layers for classifier heads with *tanh* activation loss between them. We also have used a dropout layer between them with a probability of 60%. The primary rationale for using a high dropout rate was to enhance the model's generalization ability and reduce its dependence on the training data.

4.4 Hyperparameters

For training our model, we have used AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 1e-5. We have used a batch size of 2 with gradient accumulation for 8 steps (effective batch size 16). We have used early stopping on the validation data with patience 10. Maximum document length was set to 4096 as most of the documents are large. We use the PyTorch-lightning [7] library to run the experiments and Weight & Biases [8] for logging. All of our experiments are run on NVIDIA Quadro RTX 8000 48GB.

5 Results

In this section, we report our results on subtask A and discuss our analysis. Our evaluation is based on the accuracy metric, but we have provided the micro and macro-f1 for better comparison. All the results are averaged on 3 runs with 3 different random seeds.

[7] https://lightning.ai/
[8] https://wandb.ai/

5.1 Baseline & Our Model

We use the official baseline provided by the task organizers. They have used RoBERTa-large (Liu et al., 2019) as the encoder and fine-tuned on the train data. Throughout the paper, we refer to this model as $baseline_{rob}$.

We have fine-tuned our model (shown in Fig. 1) on the training dataset. Throughout the paper, we refer to this model as $ours_{con}$

In the Table 1, we have reported the results on the official test file. $Ours_{con}$ is the final submission, and $Ours_{con}$+ is the modified version of our final model for more analysis (not official results; used for ablation study - details on §5.2). We can get a comparable result using 60% fewer parameters than the baseline. In the next section, we will see that after hyperparameter tuning, we can get around 5.7% improvement over the baseline. This supports our assumption that using a contrastive learning-based method can help machine-generated text identification.

5.2 Ablation Study

Effect of Maximum Sentence Length: The maximum sentence length is used to tokenize the document. The optimal test score is achieved with a maximum sentence length of 256. This demonstrates that the model can effectively identify machine-generated text even with documents as large as 256 words. This underscores the effectiveness and adaptability of our model's learning capabilities.

	Max Sen Length	CLS Dropout	Effective Batch Size	Macro-f1	Micro-f1	Accuracy
$Ours_{con}$	4096	0.6	16	**88.81**	**89.07**	**89.07**
$baseline_{rob}$	-	-	-	-	-	88.47
Maximum Sentence Length						
$Ours_{con}+$	128	0.6	16	88.88	89.14	89.14
$Ours_{con}+$	256			**93.30**	**93.37**	**93.36**
$Ours_{con}+$	512			88.78	89.04	89.04
$Ours_{con}+$	1024			90.99	91.13	91.13
$Ours_{con}+$	2048			91.81	91.93	91.93
$Ours_{con}$	4096			88.81	89.07	89.07
Classification Layer Dropout						
$Ours_{con}+$	4096	0	16	**92.73**	**92.81**	**92.81**
$Ours_{con}+$		0.2		90.16	90.33	90.33
$Ours_{con}+$		0.4		78.98	80.21	80.21
$Ours_{con}$		0.6		88.81	89.07	89.07
$Ours_{con}+$		0.9		82.60	83.31	83.31
Effective Batch Size						
$Ours_{con}+$	4096	0.6	2	**<u>93.80</u>**	**<u>93.86</u>**	**<u>93.86</u>**
$Ours_{con}+$			4	70.79	73.52	73.52
$Ours_{con}+$			8	76.82	78.43	78.43
$Ours_{con}$			16	88.81	89.07	89.07
$Ours_{con}+$			32	79.72	80.83	80.83
$Ours_{con}+$			64	90.64	90.80	90.80
$Ours_{con}+$			128	91.39	91.51	91.51

Table 1: Macro-f1, Micro-f1, and Accuracy score on the test result. $Ours_{con}$ - final submitted model on the shared task, $baseline_{rob}$ - official baseline model, and $Ours_{con}+$ - modified versions of our final model with more hyperparameter tuning. The **bold** value signifies the best score within a specific section, whereas the **<u>underlined</u>** value denotes the best score across all sections.

Effect of Classification Dropout: The classification dropout is applied between the two classification layers. Contrary to our initial assumption, the results presented in Table 1 indicate that using a low dropout rate (as low as 0.0) contributes positively to the model's learning process. This suggests that, even without dropout, the model's generalization to unseen data (text generated by a new model) is enabled primarily through contrastive learning and data augmentation.

Effects of (Effective) Batch Size: Due to computational constraint, we have used a fixed batch size of 2 and gradient accumulation steps of $\{1, 2, 4, 8, 16, 32, 64\}$ resulting in an effective batch size of $\{2, 4, 8, 16, 32, 64, 128\}$. From the results report on Table 1, we found that using only an effective batch size of 2 yielded superior performance compared to gradient accumulation. Notably, this configuration represents the most optimal result obtained following hyperparameter tuning, positioning us at the 8^{th} rank in the final standings. This suggests that, in this particular context, the benefits of gradient accumulation may be limited compared to simply using a smaller batch size.

6 Conclusion & Future Work

In this work, we introduce our contrastive learning-based system, which shows a comparable performance. We demonstrate that a model with half the parameters and without an ensemble of large models or hand-engineered features can show a comparable performance, which requires more exploration in this field. For future work, the use of recent prompt-based models [9] can be used for data augmentation. Also, the effect of more advanced contrastive loss, i.e., Triplet loss (Chechik et al., 2010) or InfoNCE loss (Oord et al., 2018), need to be explored.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sameer Badaskar, Sameer Badaskar, Sameer Badaskar, Sonali Agarwal, Sachin Agarwal, Shilpa Arora, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. *International Joint Conference on Natural Language Processing*.

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. Quality controlled paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 596–609, Dublin, Ireland. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28:41–75.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection. *arXiv.org*.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3).

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

[9] https://chat.openai.com/

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. 2023. Chatting and cheating: Ensuring academic integrity in the era of chatgpt. *Innovations in Education and Teaching International*, pages 1–12.

Nassim Dehouche. 2021. Plagiarism in the age of massive generative pre-trained transformers (gpt-3). *Ethics in Science and Environmental Politics*, 21:17–23.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Paul Fyfe. 2023. How to cheat on your final paper: Assigning ai for student writing. *AI & SOCIETY*, 38(4):1395–1405.

Jun Gao, Wei Wang, Changlong Yu, Huan Zhao, Wilfred Ng, and Ruifeng Xu. 2022. Improving event representation via simultaneous weakly supervised contrastive learning and clustering. *arXiv preprint arXiv:2203.07633*.

Sebastian Gehrmann, Sebastian Gehrmann, Hendrik Strobelt, Hendrik Strobelt, Gérard Rushton, Alexander M. Rush, Alexander M. Rush, and Alexander M. Rush. 2019. Gltr: Statistical detection and visualization of generated text. *Annual Meeting of the Association for Computational Linguistics*.

Jiatao Gu, Kyunghyun Cho, and Victor OK Li. 2017. Trainable greedy decoding for neural machine translation. *arXiv preprint arXiv:1702.02429*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv.org*.

Xiaotong He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *arXiv.org*.

Daphne Ippolito, Daphne Ippolito, Daniel Duckworth, Daniel Duckworth, Chris Callison-Burch, Chris Callison-Burch, Douglas Eck, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. *Annual Meeting of the Association for Computational Linguistics*.

Ganesh Jawahar, Ganesh Jawahar, Muhammad Abdul-Mageed, Muhammad Abdul-Mageed, Laks V. S. Lakshmanan, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *International Conference on Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Dongsheng Luo, Wei Cheng, Jingchao Ni, Wenchao Yu, Xuchao Zhang, Bo Zong, Yanchi Liu, Zhengzhang Chen, Dongjin Song, Haifeng Chen, et al. 2021. Unsupervised document embedding via contrastive augmentation. *arXiv preprint arXiv:2103.14542*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *International Conference on Machine Learning*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pretraining. *arXiv preprint arXiv:2201.10005*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. *IEEE Symposium on Security and Privacy*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Shubhashis Roy Dipta, Mehdi Rezaee, and Francis Ferraro. 2023. Semantically-informed hierarchical event modeling. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 353–369, Toronto, Canada. Association for Computational Linguistics.

Louis Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. *arXiv preprint arXiv:1701.03185*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro,

Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Hao Wang and Yong Dou. 2023. Sncse: contrastive learning for unsupervised sentence embedding with soft negative samples. In *International Conference on Intelligent Computing*, pages 419–431. Springer.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. SemEval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Lingling Xu, Haoran Xie, Zongxi Li, Fu Lee Wang, Weiming Wang, and Qing Li. 2023. Contrastive learning models for sentence representations. *ACM Transactions on Intelligent Systems and Technology*, 14(4):1–34.

Zhen Yang, Ming Ding, Qingsong Lv, Zhihuan Jiang, Zehai He, Yuyi Guo, Jinfeng Bai, and Jie Tang. 2023. Gpt can solve mathematical problems without a calculator. *arXiv preprint arXiv:2309.03241*.

Rowan Zellers, Rowan Zellers, Ari Holtzman, Ari Holtzman, Hannah Rashkin, Hannah Rashkin, Yonatan Bisk, Yonatan Bisk, Ali Farhadi, Ali Farhadi, Franziska Roesner, Franziska Roesner, Yejin Choi, and Yejin Choi. 2019. Defending against neural fake news. *Neural Information Processing Systems*.

Team AT at SemEval-2024 Task 8: Machine-Generated Text Detection with Semantic Embeddings

Yuchen Wei
Department of Computer Science
St. Francis Xavier University
x2020fct@stfx.ca

Abstract

This study investigates the detection of machine-generated text using several semantic embedding techniques, a critical issue in the era of advanced language models. Different methodologies were examined: GloVe embeddings, N-gram embedding models, Sentence BERT, and a concatenated embedding approach, against a fine-tuned RoBERTa baseline. The research was conducted within the framework of SemEval-2024 Task 8, encompassing tasks for binary and multi-class classification of machine-generated text.

1 Introduction

In the burgeoning field of Natural Language Processing (NLP), the distinction between human and machine-generated text is becoming an area of critical importance, particularly with the rise of advanced language models capable of producing text that closely mimics human writing. The advent of such technology poses a dual-faceted challenge: while it opens new frontiers for automation and assistance, it also necessitates robust detection mechanisms to prevent misuse and uphold information credibility. This research centers on the application of semantic embeddings to detect machine-generated text.

Semantic embeddings offer a nuanced approach to understanding and representing the meaning encapsulated within text, providing a fertile ground for discriminating between the subtleties of human and AI-authored content. This study contributes to this domain by evaluating the efficacy of various semantic embedding techniques in the context of SemEval-2024 Task 8's (Wang et al., 2024) challenges, which include the detection of machine-generated text across multiple generators and domains.

In this study, I concentrated on the application of semantic embeddings, examining and contrasting approaches such as GloVe and Sentence BERT. I developed classifiers for the task of classifying machine-generated text as part of SemEval-2024 Task 8. Specifically, my efforts were directed towards Subtask A (monolingual) and Subtask B, which involve the binary classification of machine-generated text and multi-class classification of machine-generated text, respectively.

2 Related Work

The identification and analysis of machine-generated text have become an increasingly pertinent field of study within the realm of Natural Language Processing (NLP). Previous research has primarily focused on detecting text authored by specific language models (Guo et al., 2023) or within narrow domains (Zellers et al., 2019). The latest iteration of this exploration is represented in the work by SemEval-2024 Task 8 (Wang et al., 2024), aiming at detecting text generated by a variety of models across multiple domains and languages, thus expanding the scope of investigation significantly beyond the existing literature.

Early approaches, such as those by Iyyer et al. 2014, utilized basic statistical features and machine learning models for text classification tasks, providing a foundation for subsequent research. Advancements were made by Pennington et al. 2014, who proposed a sophisticated embedding technique known as GloVe, which captures global word co-occurrence statistics (Bullinaria and Levy, 2007) to generate word representations. This technique has been widely adopted for its robustness in capturing semantic nuances.

The introduction of transformer-based (Vaswani et al., 2017) models, particularly BERT (Devlin et al., 2018) and its variants, has revolutionized the field, as demonstrated by Reimers and Gurevych 2019 with the adaptation of BERT for sentence-level embeddings (SBERT). These models have significantly outperformed traditional embeddings and

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 492–496

N-gram models in various NLP tasks due to their deep contextual understanding and adaptability to different tasks and domains. Moreover, RoBERTa (Liu et al., 2019) (A Robustly Optimized BERT Pretraining Approach) refines the BERT model's training methodology to substantially improve performance across a spectrum of NLP benchmarks.

More recently, Large Language Models (LLMs) have revolutionized text generation, achieving human-like proficiency across diverse writing tasks. As LLMs like ChatGPT (Brown et al., 2020) and its successors become more adept at generating coherent and contextually relevant narratives, the importance of distinguishing between machine-generated and human-produced text grows, primarily to ensure transparency and mitigate the spread of misinformation. Consequently, developing robust detection methods for machine-generated text is crucial in maintaining the integrity of information and upholding trust in digital communications.

3 Methods

In this study, I explored four semantic embedding methods to evaluate against the fine-tuned RoBERTa baseline provided by the task coordinators (Wang et al., 2024). The methods employed encompass the GloVe (Pennington et al., 2014) embedding method, the training N-gram embedding method, Sentence BERT method, and the concatenated embedding method. In this section, I will present the methodologies applied to address Subtask A (monolingual) and Subtask B. Their primary distinction lies in the extraction of text features.

3.1 GloVe Embedding Method

Pre-trained GloVe embeddings are a set of vector representations for words that have been previously trained on large corpora, encapsulating rich semantic and syntactic relationships between words. In this approach, for each piece of text, GloVe embeddings were utilized to derive the text feature, calculated as the mean of the GloVe embeddings for each word within the text. Subsequently, a straightforward fully connected neural network, comprising several hidden layers, was constructed to perform classification.

I experimented with GloVe embeddings of varying dimensions (100d, 200d, 300d) and employed Smooth Inverse Frequency (SIF) weighted averaging as the method for averaging. This approach (Arora et al., 2017) has been demonstrated to enhance the performance of text embedding usage.

3.2 Training N-gram Embedding Method

In addition to GloVe embeddings, I explored the training of word embeddings through an N-gram neural network model. This model was designed to train a word embedding layer with the objective of predicting the subsequent word based on a given sequence of N words. Subsequently, the trained word embeddings were utilized to extract text embeddings, which then served as the basis for classification, similar to the methodology applied with GloVe embeddings.

3.3 Sentence BERT Method

Sentence BERT (Reimers and Gurevych, 2019) is a modification of the pre-trained BERT model that enhances its capabilities for generating sentence-level embeddings, facilitating more efficient and semantically meaningful comparisons between sentences. In this approach, similar to others, classification is conducted through a fully connected neural network; however, Sentence BERT is employed for the extraction of text features.

3.4 Concatenated Embedding Method

In this methodology, I concatenated word embeddings with Sentence BERT embeddings to serve as the text feature embeddings. The objective is to leverage the strengths of both approaches to enhance classification performance. The dimension of the concatenated embedding for each sample's text equals to the sum of the dimensions of the word embeddings and the SBERT embeddings. I experimented with combining GloVe and SBERT, as well as N-gram embeddings with SBERT. Ultimately, in a similar vein, a fully connected neural network was employed for inputting concatenated embeddings and performing the classification tasks.

4 Dataset and Experimental Setting

4.1 Dataset

The coordinators of SemEval-2024 Task 8 have introduced three subtasks focused on the detection of machine-generated text, encompassing multi-generator, multi-domain, and multi-lingual challenges. The first task (Subtask A) is framed as a binary classification challenge, with the goal being to differentiate between human-written and machine-generated text. Subtask A is divided into two segments: monolingual and multilingual. The mono-

lingual segment contains 119,757 training samples, while the multilingual segment includes 172,417 training samples. In this research, my attention is solely directed towards the monolingual task, which exclusively involves texts in English. Its training set comprises 56,406 samples generated by machines and 63,351 samples authored by humans.

The second task (Subtask B) is structured as a multi-class classification challenge, wherein the labels for text samples encompass *human*, *ChatGPT*, *Cohere*, *Davinci*, *Bloomz*, and *Dolly*. This task requires classifiers to not merely determine whether a given text is machine-generated but also to identify the specific type of language model (e.g., ChatGPT (Brown et al., 2020), Cohere (Cohere Technologies, 2021)) responsible for its generation. This task encompasses a training set comprising 71,027 samples (11,997 samples for *human*, 11,995 samples for *ChatGPT*, 11,336 samples for *Cohere*, 11,999 samples for *Davinci*, 11,998 samples for *Bloomz*, 11,702 samples for *Dolly*).

The third task (Subtask C) focuses on locating the boundary within each mixed text sample. For this subtask, the provided samples are mixed texts, consisting of a human-written segment followed by a machine-generated segment. The primary objective is to identify the transition point between these two segments. This subtask includes 3,649 training samples. In my research, I did not engage with this particular subtask.

4.2 Experimental Setting

In this study, I applied my methodologies to Subtask A and Subtask B, assessing their effectiveness on the training sets using a K-fold cross-validation approach with K=5, as well as on the testing sets. Accuracy was selected as the evaluation metric for this analysis. I employed a fine-tuned RoBERTa model (Liu et al., 2019) as the baseline against which to compare my approaches. The released testing sets for Subtask A (monolingual) and Subtask B consist of 34,272 and 18,000 samples, respectively. Within the Subtask A testing set, there are 18,000 machine-generated samples and 16,272 human-written samples. For Subtask B's testing set, each label is represented by 3,000 samples.

5 Experimental Results

In this section, I will present and analyze the experimental outcomes derived from the implementation of my methodologies.

5.1 GloVe Embedding Method Results

The data in Table 1 elucidates the efficacy of the GloVe embedding methodology when applied to Subtask A (binary classification) and Subtask B (multi-class classification) of text classification. The results are segmented according to the dimensionalities of the GloVe embeddings—100, 200, and 300—and benchmarked against the performance of a fine-tuned RoBERTa model. A pattern of ascending accuracy aligns with the increase in GloVe dimensions for Subtask A, culminating in a maximum accuracy of 62.1% on the test set for the 300-dimensional GloVe model. Conversely, for Subtask B, the trend, though similar, is subdued, with the 300-dimensional model attaining an accuracy of 34.6% on the test set. The RoBERTa model, which serves as the baseline, outshines the GloVe models with a substantial margin, exhibiting peak accuracies of 73.6% on Subtask A and 48.6% on Subtask B during test evaluations.

It's clear that the dimensionality of GloVe embeddings has a direct correlation with the accuracy of the models; higher dimensions lead to more expressive embeddings and, consequently, better performance. However, despite the improvements seen with 300-dimensional embeddings, the GloVe models fall short when compared to the fine-tuned RoBERTa model.

Method Name	A K-fold	A Test	B K-fold	B Test
GloVe 100d	75.6%	59.6%	46.5%	31.3%
GloVe 200d	78.2%	61.4%	47.9%	32.9%
GloVe 300d	79.7%	62.1%	49.3%	34.6%
RoBERTa	93.8%	73.6%	63.1%	48.6%

Table 1: The experimental outcomes for the GloVe embedding method, spanning various dimensions.

5.2 Training N-gram Embedding Method Results

Table 2 presents the performance of the N-gram embedding method for both Subtask A and Subtask B, showing a progression in accuracy as the value of N increases, indicating that loner contexts inputed by N-grams contribute to more accurate models. Specifically, for Subtask A, the 2-gram model starts with a K-fold accuracy of 79.1% and a test accuracy of 60.3%, which gradually increases with the 5-gram model reaching a K-fold accuracy of 82.1% and a test accuracy of 61.4%. For Subtask B, the increase in N-gram size also correlates

with a slight increase in accuracy, with the 5-gram model achieving a K-fold accuracy of 48.7% and a test accuracy of 33.9%.

When compared to the GloVe embedding method from the earlier table, the N-gram models demonstrate a competitive edge in K-fold accuracy for Subtask A, but this edge diminishes in the test results where GloVe 300d outperforms the N-gram methods. For Subtask B, the N-gram models show a similar pattern with slightly better performance compared to the GloVe 100d and 200d models but are still outperformed by the GloVe 300d and the fine-tuned RoBERTa model. RoBERTa continues to maintain a significant lead over both GloVe and N-gram methods, underscoring the effectiveness of contextualized embeddings over both static and N-gram embeddings for the tasks at hand.

Method Name	A K-fold	A Test	B K-fold	B Test
2-gram	79.1%	60.3%	47.9%	31.5%
3-gram	81.4%	61.4%	48.7%	33.3%
4-gram	80.5%	61.7%	49.3%	33.2%
5-gram	82.1%	61.4	48.7%	33.9%
RoBERTa	93.8%	73.6%	63.1%	48.6%

Table 2: The experimental results for the N-gram embedding method, across different values of N representing the number of words in the input context.

5.3 Sentence BERT Method Results

Table 3 illustrates the performance for the Sentence BERT (SBERT) method applied to Subtask A and Subtask B, with the variation in performance attributed to the different counts of hidden layers, denoted as H. The results reveal that for Subtask A, the model with one hidden layer (SBERT H=1) achieved a K-fold accuracy of 84.1% and a test accuracy of 66.3%. As the number of hidden layers increased, there was a marginal improvement in K-fold accuracy, peaking at 83.6% for four hidden layers (SBERT H=4), while the test accuracy remained relatively stable, peaking at 66.3% for one hidden layer. In Subtask B, the trend is less clear, with SBERT H=1 achieving the highest test accuracy at 38.1%, despite having a lower K-fold accuracy compared to models with more hidden layers. When compared to the GloVe method and the N-gram embedding method from previous tables, SBERT tends to offer improved test accuracy in both Subtask A and Subtask B.

Method Name	A K-fold	A Test	B K-fold	B Test
SBERT H=1	84.1%	66.3%	52.6%	38.1%
SBERT H=2	82.1%	65.6%	52.7%	37.5%
SBERT H=3	82.7%	65.8%	51.5%	36.7%
SBERT H=4	83.6%	65.8%	50.7%	37.1%
RoBERTa	93.8%	73.6%	63.1%	48.6%

Table 3: The experimental results for the Sentence BERT (SBERT) method, varying across different counts of H, which denotes the number of hidden layers.

5.4 Concatenated Embedding Method Results

Table 4 displays the experimental results for the concatenated embedding method, combining Sentence BERT (SBERT) with GloVe embeddings and a 5-gram model. The SBERT+GloVe model exhibits a K-fold accuracy of 85.4% and a test accuracy of 68.1% for Subtask A, while for Subtask B, it shows a K-fold accuracy of 53.2% and a test accuracy of 38.9%. The SBERT+5-gram model slightly outperforms the SBERT+GloVe in Subtask A with a K-fold accuracy of 86.7% and a test accuracy of 67.3%, and a K-fold accuracy of 54.1% for Subtask B, though the test accuracy is slightly lower at 38.3%. These results indicate that combining SBERT with 5-gram embeddings or GloVe embeddings could provide a marginal improvement over methods that apply each exclusively.

Method Name	A K-fold	A Test	B K-fold	B Test
SBERT+GloVe	85.4%	68.1%	53.2%	38.9%
SBERT+5-gram	86.7%	67.3%	54.1%	38.3%
RoBERTa	93.8%	73.6%	63.1%	48.6%

Table 4: The experimental results for the concatenated embedding method.

5.5 Competition Submission

Since my methods did not perform as well as the fine-tuned RoBERTa, I ultimately submitted the predictions of my fine-tuned RoBERTa model on the test sets for Subtask A and Subtask B. In the end, my predictions ranked 79th for Subtask A and 63rd for Subtask B.

6 Conclusion

In conclusion, this research has contributed to the field of detecting machine-generated text by exploring the efficacy of various semantic embedding methodologies. The results present the performance of several pre-trained semantic embeddings, like GloVe and SBERT, on the tasks of machine-generated text detection in SemEval-2024 Task 8.

Due to the superior performance of the fine-tuned RoBERTa model over all my methods I implemented, I ultimately chose to submit the prediction results obtained from RoBERTa for the SemEval competition.

7 Limitation

The limitations of this work mainly exist in two aspects. First, the methods used are too traditional and outdated. Secondly, its performance is not as good as the baseline.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39:510–526.

Cohere Technologies. 2021. Cohere natural language processing api. https://cohere.ai. Accessed: 2024-02-17.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 633–644.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

JN666 at SemEval-2024 Task 7: NumEval: Numeral-Aware Language Understanding and Generation

Xinyi Liu, Xintong Liu and **Hengyang Lu**
School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China
1130174701@qq.com
liuxintong@stu.jiangnan.edu.cn
luhengyang@jiangnan.edu.cn

Abstract

This paper is submitted for SemEval-2027 task 7: Enhancing the Model's Understanding and Generation of Numerical Values. The dataset for this task is NQuAD [1], which requires us to select the most suitable option number from four numerical options to fill in the blank in a news article based on the context. Based on the BertForMultipleChoice model, we proposed two new models, MC BERT and SSC BERT, and improved the model's numerical understanding ability by pre-training the model on numerical comparison tasks. Ultimately, our best-performing model achieved an accuracy rate of 79.40%, which is 9.45% higher than the accuracy rate of NEMo [1].

1 Introduction

In the field of Natural Language Processing (NLP), the understanding and analysis of textual data have always been the main focus. However, the numerical information in these textual data is often overlooked. Numerals play a significant role in our daily life and work, providing rich information such as dates, times, quantities, proportions, and money. Although numerals may not occupy a large proportion in the text, their existence is crucial for understanding the meaning of the text.

To better evaluate the numerical understanding ability of models, [1] proposed the Numeral-related Question Answering Dataset (NQuAD) [1], which is specifically designed to evaluate and enhance the model's ability in Reading Comprehension of the Numerals in Text. In our work, our goal is to improve the performance of the BERT model on the NQuAD [1]. For this purpose, we propose a new method that can effectively handle numerical information. Our experimental results show that our method has achieved significant performance improvement on the NQuAD [1]. This proves the effectiveness of our method in handling numerical information and also shows the potential of our method in enhancing the numerical understanding ability of the model.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 introduces the dataset and tasks. Section 4 describes our method in detail. Section 5 reports our experimental results and analysis. Finally, we conclude our work.

2 Related Work

In recent years, pre-training tasks have become increasingly prevalent in the field of Natural Language Processing (NLP). The design goal of pre-training tasks is to enhance the model's understanding of natural language through learning from a large amount of unlabeled data, thereby improving the model's performance on specific tasks in the subsequent fine-tuning stage. In this trend, [2] shows that by pre-training on numerical comparison tasks, the model's understanding of numerals can be significantly improved. The model, after pre-training, also shows a significant performance improvement on other numeral-related tasks.

Multiple choice [3] format represents a category within machine reading comprehension tasks. In this context, [4] proposed a Multi-stage Multi-task learning framework (MMM) aimed at enhancing the performance of multiple choice tasks, [5]introduced the Dual Co-Matching Network (DCMN+), a model that emulates human problem-solving strategies, and [6] presents a two-step strategy for enhancing the performance of Large Language Models (LLMs) on multiple choice tasks.

Recent research has increasingly recognized the significance of numerical data within text. The NQuAD [1] dataset has been instrumental in examining the relationship between numerical values in news headlines and the corresponding figures within the articles. Similarly, the FNXL [7] dataset has brought attention to the numerical data con-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 497–502

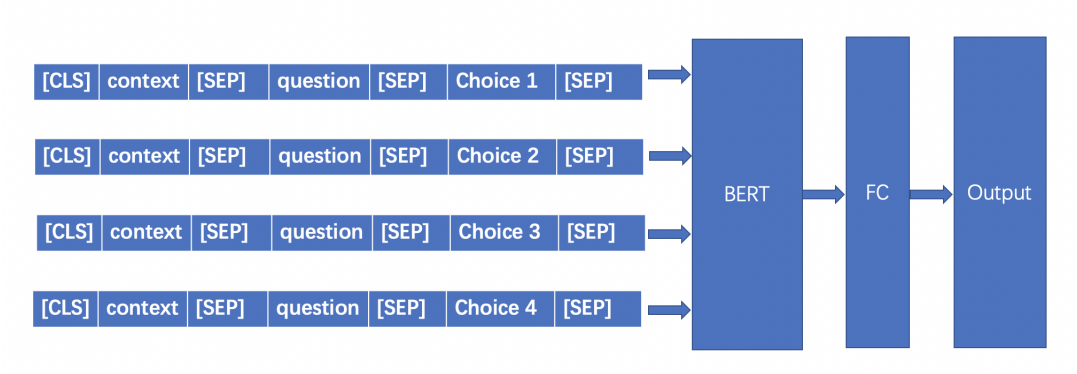


Figure 1: **Architecture of BertForMultipleChoice model**

tained within the periodic financial reports submitted by publicly listed companies.

The BertForMultipleChoice model is a variant of the BERT [8], specifically designed for multiple-choice tasks. Figure 1 is the architecture of BertForMultipleChoice, it adds a multiple-choice classification head on top of the BERT model, mainly for handling multiple-choice tasks. The input to the BertForMultipleChoice model includes a question and multiple potential answers (options), and the goal of the model is to select the most reasonable answer. This model has shown excellent performance in tasks that require choosing one answer from multiple options, such as reading comprehension and sentiment analysis.

3 Dataset and Tasks

Figure 2 shows an example in NQuAD [1] dataset, including a news article, a question stem and four answer options. Our task is predicting the correct the option.

NQuAD [1] collects news articles from the data vendor, MoneyDJ1, and get the news articles within the period from June 22, 2013 to June 20, 2018. A total of 75,448 Chinese news articles are collected. A total of 43,787 news articles are selected, and 46.97% of the headlines contain more than one numeral. The average number of numerals in the headline and in the content are 1.65 and 29.48, respectively. Each numeral in each headline is used to form a question, thus NQuAD [1] dataset finally obtain 71,998 questions and separate 80% of the instances as the training set and the rest of the instances form a test set.

News Article:
Major banks take the lead in self-discipline. The five major banks' newly imposed mortgage interest rates climbed to **1.986**% in May. ... Also approaching **2**% integer alert ... Up to **2.5**% ... Also increased by **0.04** percentage points from the previous month ... Prevent the housing market bubble from fully starting.

Question Stem: Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly___%.

Answer Options: (A) 0.04 (B) 1.986 (C) 2 (D) 2.5

Answer: (C)

Figure 2: **An example question in NQuAD.**

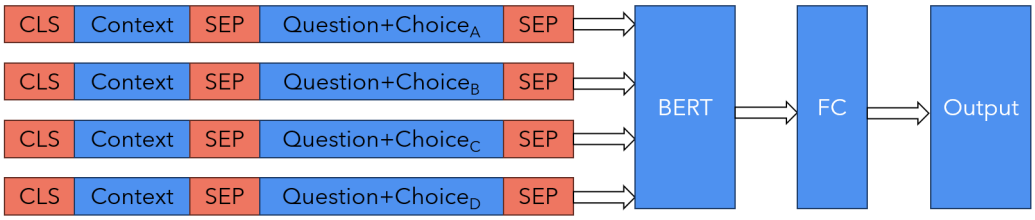


Figure 3: **Architecture of the proposed model, MC BERT**

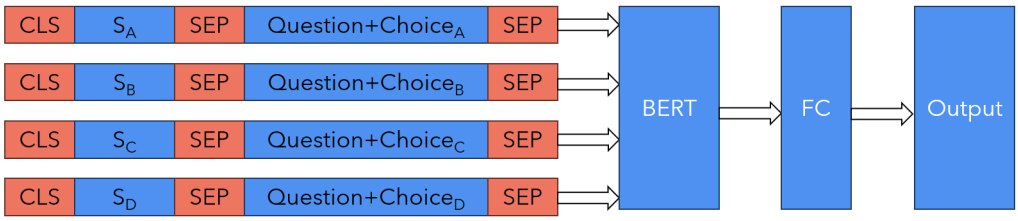


Figure 4: **Architecture of the proposed model, SSC BERT**

MC BERT Input:
[[CLS]Also increased by **0.04** percentage points from the previous month;he five major banks' newly-imposed mortgage interest rates climbed to **1.986**% in May.;Also approaching **2**% integer alert;Up to **2.5**[SEP]Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **0.04**%.[SEP],
[CLS]Also increased by **0.04** percentage points from the previous month;he five major banks' newly-imposed mortgage interest rates climbed to **1.986**% in May.;Also approaching **2**% integer alert;Up to **2.5**[SEP]Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **1.986**%.[SEP],
[CLS]Also increased by **0.04** percentage points from the previous month;he five major banks' newly-imposed mortgage interest rates climbed to **1.986**% in May.;Also approaching **2**% integer alert;Up to **2.5**[SEP]Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **2**%.[SEP],
[CLS]Also increased by **0.04** percentage points from the previous month;he five major banks' newly-imposed mortgage interest rates climbed to **1.986**% in May.;Also approaching **2**% integer alert;Up to **2.5**[SEP]Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **2.5**%.[SEP]]

SSC BERT Input:
[[CLS]Also increased by **0.04** percentage points from the previous month[SEP]Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **0.04**%.[SEP],
[CLS]he five major banks' newly-imposed mortgage interest rates climbed to **1.986**% in May.[SEP]Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **1.986**%.[SEP],
[CLS]Also approaching **2**% integer alert[SEP]Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **2**%.[SEP],
[CLS]Up to **2.5**[SEP]Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **2.5**%.[SEP]]

Figure 5: **Examples of MC BERT Input and SSC BERT Input.**

4 Methods

4.1 Multiple Choice

Multiple Choice [3] format represents a category within machine reading comprehension tasks. Our present endeavor aligns with this format. The BertForMultipleChoice model, an adaptation of the BERT [8] framework, is expressly engineered to process tasks involving this format. So we choose the BertForMultipleChoice model as our baseline model.

Resent research [5] suggests that existing multi-choice MRC models learn the passage representation with all the sentences in one-shot, which is inefficient and counter-intuitive. Their research indicates that the model should be extremely beneficial if it focuses on a few key evidence sentences. At the same time, the "sentences_containing_the_numeral_in_answer_options" in the NQuAD [1] dataset are four sentences that contain each of the options. So, our strategy is using this sentences as the context of inputs. In the BertForMultipleChoice model, we connect this sentences as the context, and the Question Stem and Answer Options are used as the question and Choices. After that, we find that humans often employ the method of substituting potential solutions into the given problem in the context of problem-solving. Be inspired by this, we fill in the blanks of the question with the options. As shown in Figure 3, we named this new model MC BERT. Furthermore, when humans solve problems, they will finally compare the question and options with key sentences one by one, and then choose the most matching option. So we further propose an improvement strategy, that is, changing the context to the sentence corresponding to each option to increase differentiation. As shown in Figure 4, we named this improved model SSC BERT. Here are examples of input:

- Context:Also increased by **0.04** percentage points from the previous month,he five major banks' newly-imposed mortgage interest rates climbed to **1.986**% in May.,Also approaching **2**% integer alert,Up to **2.5**%
- S_A: Also increased by **0.04** percentage points from the previous month
- S_B: he five major banks' newly-imposed mortgage interest rates climbed to **1.986**% in May.
- S_C: Also approaching **2**% integer alert
- S_D: Up to **2.5**%
- Question: Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly ___%.
- $QC_A/QC_B/QC_C/QC_D$: Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly **0.04/1.986/2/2.5**%.

Figure 5 are final input examples of MC BERT and SSC BERT.

4.2 Pre-training

Chapter 2 mentioned that pre-training models on simple numerical comparison tasks can enhance the model's numerical understanding. Therefore, we pre-train 'bert-base-chinese' on numerical comparison tasks according to the method mentioned in the [2]. The pre-training model is then trained according to the three frameworks mentioned above. We finally obtain six models: BertForMultipleChoice, MC BERT, SSC BERT, PreBertForMultipleChoice, Pre-MC BERT and Pre-SSC BERT.

4.3 Implementation Details

Trained on the ComNum dataset [2]. Given that NQuAD [1] is a Chinese dataset, we first replaced the English in the ComNum dataset [2] with the corresponding Chinese. The pre-trained model we used is bert-base-chinese, the maximum sequence length is 32, batch size is 32. The loss function employed is the cross-entropy loss, with the AdamW optimizer. The learning rate is set at 5e-6, and epsilon is 1e-8.

Trained on the NQuAD dataset [1]. We employed the BertForMultipleChoice from the transformers library to train the model and divided the training set into a training set and a validation set at a ratio of 4:1. The maximum sequence length is 128, batch size is 32. The loss function employed is the cross-entropy loss, with the AdamW optimizer. The learning rate is set at 5e-6 (If the model used was trained on the ComNum dataset [2], the learning rate is set at 5e-5), and epsilon is 1e-8. After applying softmax function to the model's output, we selected the index with the highest probability as the output.

Model	Accuracy
BERT Embedding Similarity	57.30%
Vanilla BERT	66.41%
BERT-BiGRU	67.15%
BERT-CNN	63.92%
NEMo	69.95%
BertForMultipleChoice	74.48%
MC BERT(ours)	76.83%
SSC BERT(ours)	78.21%
Pre-BertForMultipleChoice	74.91%
Pre-MC BERT(ours)	77.16%
Pre-SSC BERT(ours)	**79.40**%

Table 1: Experimental results.

5 Result

As shown in Table 1, the accuracy of our six models all exceed NEMo [1], which indicates that the BertForMultipleChoice model framework can effectively handle and solve the specific requirements and challenges of this task. Among the three models obtained by directly training with bert-base-chinese, the accuracy of MC BERT is higher than that of BertForMultipleChoice, indicating that directly filling in the blanks of the question with options is effective. Furthermore, the accuracy of SSC BERT surpasses that of MC BERT, suggesting that comparing the question and options with key sentences individually can also enhance the model's performance. Meanwhile, The accuracy of the three models trained using pre-trained models are higher than that of the models obtained by directly training with bert-base-chinese, which further confirms the conclusion that in some simple numerical related tasks, pre-trained models can enhance the model's numerical understanding ability [2].

6 Conclusion

In this work, we select the BertForMultiple model as the baseline model and propose two new models based on it: MC BERT and SSC BERT. The accuracy of these models all exceeded the results of NEMo [1], which confirmed the applicability of the BertForMultiple model framework for our task, the results also indicate that the method of inserting options into the blanks of the question and individually comparing them with key sentences is effective. We also refer to the method in reference [2], pre-trained the bert-base-chinese on numerical comparison task, and use the pre-trained model for subsequent training. The experimental results show that this method can improve the model's numerical understanding ability, thereby making the model perform better on numerical related tasks. Finally, we found that the Pre-SSC BERT model had the highest accuracy, and its accuracy was 9.45% higher than the NEMo model [1], which further proved the effectiveness of our method.In summary, our research proposes a new model training and optimization strategy, which has been proven to be effective and superior in experiments.

However, in analyzing the cases where our model made incorrect judgments, we identify some shortcomings. We notice that humans can easily understand the equivalence between different descriptions of the same thing, but the model cannot. For example, our model cannot understand that '%' and 'Cheng'('into')are equivalent, 'EPS' and 'Earning Per Share' are equivalent, and it cannot understand that 'January' is 'Q1'. To address this issue, we tried to add some specific examples to the dataset of numerical comparison tasks, such as '10% is equal to 1 Cheng' and 'EPS 100 is equal to Earning Per Share 100', hoping that the model could understand the equivalence between two different descriptions of things through this method. However, the final result did not meet our expectations, indicating that our model still needs improvement in handling these types of problems. We will continue to explore this issue in future research, with the aim of improving the model's understanding ability and accuracy.

7 ACKNOWLEDGMENTS

This research was funded by the China Postdoctoral Science Foundation (Grant No. 2022M711360), in part by the Laboratory for Advanced Computing and Intelligence Engineering.

References

[1] Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. "NQuAD: 70,000+ Questions for Machine Comprehension of the Numerals in Text". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. CIKM '21. Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 2925–2929. ISBN: 9781450384469. DOI: 10.1145/

3459637.3482155. URL: https://doi.org/10.1145/3459637.3482155.

[2] Chung-Chi Chen et al. "Improving Numeracy by Input Reframing and Quantitative Pre-Finetuning Task". In: *Findings of the Association for Computational Linguistics: EACL 2023*. Ed. by Andreas Vlachos and Isabelle Augenstein. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 69–77. DOI: 10.18653/v1/2023.findings-eacl.4. URL: https://aclanthology.org/2023.findings-eacl.4.

[3] Shanshan Liu et al. "Neural Machine Reading Comprehension: Methods And Trends". In: *APPLIED SCIENCES-BASEL* 9.18 (2019).

[4] Di Jin et al. "Mmm: Multi-Stage Multi-Task Learning For Multi-Choice Reading Comprehension". In: *National Conference on Artificial Intelligence* 34.05 (2020), pp. 8010–8017.

[5] Shuiliang Zhang et al. "DCMN+: Dual Co-Matching Network for Multi-Choice Reading Comprehension." In: *THIRTY-FOURTH AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, THE THIRTY-SECOND INNOVATIVE APPLICATIONS OF ARTIFICIAL INTELLIGENCE CONFERENCE AND THE TENTH AAAI SYMPOSIUM ON EDUCATIONAL ADVANCES IN ARTIFICIAL INTELLIGENCE* 34.05 (2020), pp. 9563–9570.

[6] Chenkai Ma and Xinya Du. "POE: Process of Elimination for Multiple Choice Reasoning". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 4487–4496. DOI: 10.18653/v1/2023.emnlp-main.273. URL: https://aclanthology.org/2023.emnlp-main.273.

[7] Soumya Sharma et al. "Financial Numeric Extreme Labelling: A dataset and benchmarking". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 3550–3561. DOI: 10.18653/v1/2023.findings-acl.219. URL: https://aclanthology.org/2023.findings-acl.219.

[8] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

BERTastic at SemEval-2024 Task 4: State-of-the-Art Multilingual Propaganda Detection in Memes via Zero-Shot Learning with Vision-Language Models

Tarek Mahmoud[†◇], **Preslav Nakov**[†]
[†]Mohamed Bin Zayed University of Artificial Intelligence,[◇] Presight
{tarek.mahmoud,preslav.nakov}@mbzuai.ac.ae

Abstract

Analyzing propagandistic memes in a multilingual, multimodal dataset is a challenging problem due to the inherent complexity of memes' multimodal content, which combines images, text, and often, nuanced context. In this paper, we use a VLM in a zero-shot approach to detect propagandistic memes and achieve a state-of-the-art average macro F1 of $\mathbf{66.7}\%$ over all languages. Notably, we outperform other systems on North Macedonian memes, and obtain competitive results on Bulgarian and Arabic memes. We also present our early fusion approach for identifying persuasion techniques in memes in a hierarchical multilabel classification setting. This approach outperforms all other approaches in average hierarchical precision with an average score of $\mathbf{77.66}\%$. The systems presented contribute to the evolving field of research on the detection of persuasion techniques in multimodal datasets by offering insights that could be of use in the development of more effective tools for combating online propaganda.

1 Introduction

Propaganda is an ancient technique that has existed for thousands of years[1]. The way propaganda is understood today was formalized between 1937 and 1942 by the Institute of Propaganda Analysis through a series of publications (Cantril (1938), Edwards (1938), Lavine et al. (1940), and Brace (1939)). Britannica defines propaganda as the "dissemination of information—facts, arguments, rumours, half-truths, or lies—to influence public opinion."[1]. Propaganda can be beneficial when it unites people behind a noble or beneficial cause. It can also be harmful if it leads to tensions, destabilization, and the death of millions. In our digitally mediated world, transmitting (dis)information to millions of people occurs in seconds. Hence, the adverse effects of propaganda are accelerated and amplified. Propaganda has been used to influence public opinion on Brexit (Rawlinson, 2020), US elections (Chernobrov and Briant, 2020), and the Ukraine crisis (Chernobrov and Briant, 2020). Thus, it is easy to see the damaging effects propaganda has already caused and continues to inflict.

Propaganda takes many forms. It could be broadcast on television (Pan et al., 2020), spread through coordinated communities on social media (Hristakieva et al., 2022), transmitted across national borders through loudspeakers (Seo, 2018), disseminated via news articles (Nakov et al., 2022), published on blogs (Burgers, 2017), or could even exist on postage stamps (Lauritzen, 1988). More recently, memes have become powerful tools for the dissemination of political messages. The visual and textual simplicity of memes, combined with their viral nature, allows them to be rapidly consumed and shared across social media platforms, reaching vast audiences with minimal effort. This level of accessibility makes memes an attractive medium for propagandists seeking to subtly influence public opinion, disseminate misinformation, and polarize communities.

Consequently, there has been an increased need for and interest in propaganda identification in the research community. The most difficult challenge is that propaganda is often based on kernels of truth and is presented in a misleading way, making it seem genuine. Hence, training a model to detect propaganda is challenging, given the subtlety in how propaganda masquerades as an ordinary text or an innocent, funny meme. Moreover, interpreting any such model's results could also be problematic. With memes, the challenge, is further exacerbated due to the inherent complexity of memes' multimodal content, which combines images, text, and often, nuanced context, making the detection of propaganda all the more challeng-

[1] https://www.britannica.com/topic/propaganda

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 503–510

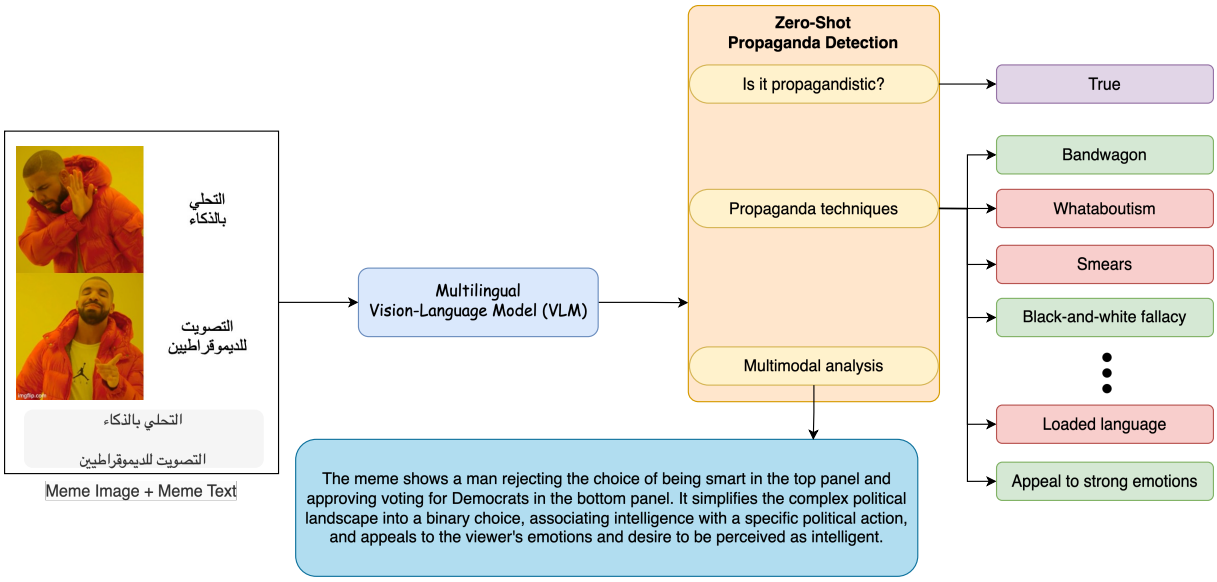


Figure 1: **Zero-shot propaganda detection overview:** This instance shows an Arabic meme that reads "being smart" in the top half and "voting democrat" in the other half. The figure illustrates the comprehensive analysis of a meme and the fine-grained output obtained from a VLM using zero-shot learning. The model accepts as input both the meme image and the meme text, and is prompted to provide three outputs. It provides a multimodal analysis and description of the meme. It also identifies which technique out of 22 possible techniques are present in the meme. Finally, it makes a determination whether the meme is propagandistic or not.

ing. Furthermore, the language used in memes is characteristically concise, often consisting of mere sentence fragments or a few keywords. Consequently, developing systems that consider only the textual content in isolation from the accompanying image presents a significant challenge.

The model explainability challenge has been tackled by Da San Martino et al. (2019) for texts and by Dimitrov et al. (2021) for memes via the introduction of fine-grained propaganda detection tasks and datasets. The tasks required identifying the propaganda technique(s) out of over eighteen techniques in a multi-label classification formulation. This fine level of granularity increases interpretability of propaganda detection models. However, the subtlety challenge is still prevalent. In addition to the subtlety challenge, propaganda detection research is scarce on multimodal datasets in general, and on memes in particular, correlating with a scarcity of datasets. Moreover, the majority of research and datasets are monolingual and consider mainly the English language. Therefore, the difficulty of tackling propaganda in memes also extends to include the scarcity of multilingual meme datasets. There are recent efforts to address this challenge, which are currently spearheaded by the shared task on *Multilingual Detection of Persuasion Techniques in Memes* (Dimitrov et al., 2024) described in detail in 2.1.

In this paper, we capitalize on recent advances in Vision Language Models (VLMs) (Zhang et al., 2023) and Pretrained Language Models (PLMs) (Zhao et al., 2023), and highlight the following contributions:

- We achieve state-of-the-art performance on multilingual, multimodal propaganda detection in memes with an average macro F1 score of $\mathbf{66.7}\%$ using a zero-shot VLM approach (see Figure 1). This includes state-of-the-art performance on North Macedonian memes, and competitive results on Bulgarian and Arabic memes.

- We present a early fusion approach for identifying persuasion techniques in memes in a hierarchical multilabel classification setting. This approach outperforms all other approaches in average hierarchical precision with an average score of $\mathbf{77.66}\%$.

2 Background

2.1 Task Formulation

The "Multilingual Detection of Persuasion Techniques in Memes" task contains three subtasks addressing the challenge of identifying persuasion

Subtask	Binary		Multilabel	
Split	Train	Test	Train	Test
English	1650	600	9050	1500
Arabic	-	160	-	120
Bulgarian	-	100	-	436
North Macedonian	-	100	-	259

Table 1: Dataset summary. All labeled data from the training, development, and validation sets are merged and included under the training split. We also augment the multilabel training split with non-propagandistic samples from the binary training split.

techniques used in memes out of which we describe two subtasks. One subtask simplifies the challenge to a binary classification task, determining the presence or absence of any persuasion technique in a meme. In more concrete terms, given a text-image pair $p = (m, t)$ where m is the meme image and t is the meme text, the goal is to predict whether p is propagandistic or not.

The other subtask requires the identification of one or more of twenty-two persuasion techniques within a meme. That is, given a text-image pair $p_i = (m, t)$, the goal is to learn a mapping $f : p \rightarrow K$ where $K = [k_1, ..., k_n]$ and $k_j \in \{True, False\}$ denotes whether p_i contains the j^{th} persuasion technique and n denotes the total number of persuasion labels, which is 22 in this subtask.

The task employs macro-F1 scores for binary classification, and hierarchical-F1 scores for multilabel classification.

Table 1 summarizes the dataset. Note that the task introduces memes in languages other than English without any labels in order to evaluate the models' zero-shot learning capabilities. Figure 2 analyzes how balanced the label distribution is in the dataset of the multilabel task. It is clear the dataset is highly imbalanced. The binary task's dataset is also imbalanced with two-thirds of the training data being propagandistic.

2.2 Related Work

Recent advancements have highlighted the multimodal nature of modern propaganda, particularly within social media. The integration of text and visual content in memes presents a unique challenge for detection algorithms and models. Recognizing this, Dimitrov et al. (2021) introduce a multi-label multimodal task focused on identifying the specific propaganda techniques used in memes. The authors have compiled and released a corpus of approximately one thousand memes. This collection is annotated with twenty two distinct propaganda techniques. These techniques appear either in the textual content, the image content, or a combination of both. The creation of such a dataset is a significant contribution to the field, providing a foundational resource for developing and evaluating propaganda analysis models on memes. One limitation of this dataset is that it only contains English memes. This challenge is overcome by Dimitrov et al. (2024) who introduce a multilingual meme dataset that contains approximately ten thousand memes in four languages including English, Arabic, Bulgarian, and North Macedonian. In this study, we use this dataset.

Dimitrov et al. (2021) evaluate many baselines on the English meme dataset. These approaches include text models, image models, and multimodal models. Unlike our work, none of these baselines utilize a zero-shot learning approach using VLMs.

3 System Overview

3.1 Zero-Shot Detection of Propagandistic Memes using Vision-Language Models

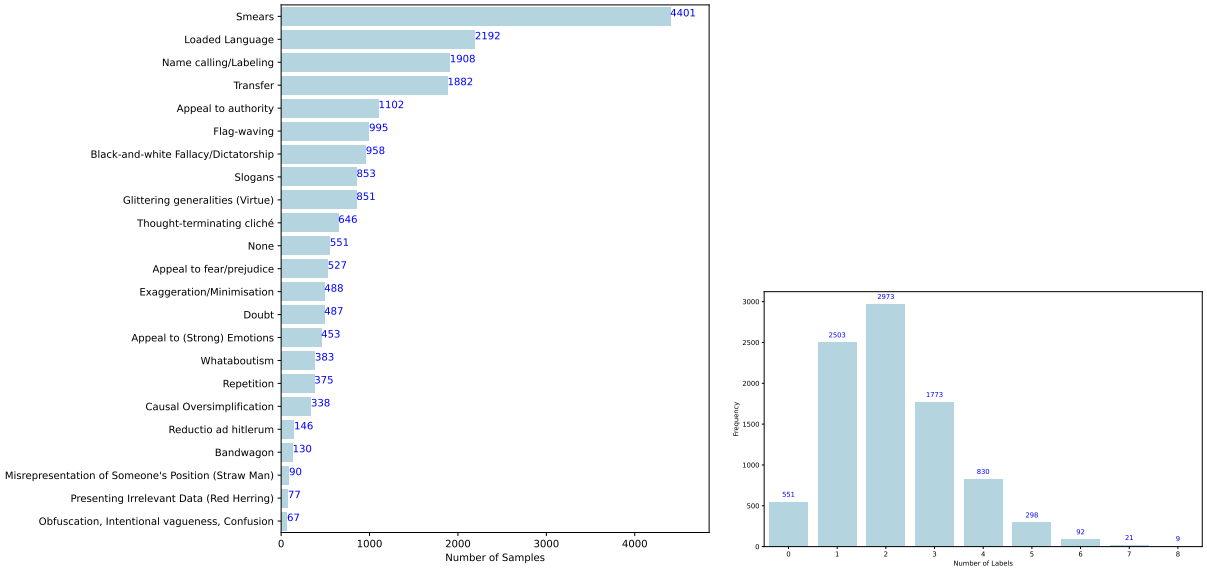
We employ zero-shot detection of propagandistic memes using GPT-4V (OpenAI, 2023). The core objective of our system illustrated in Figures 1 and 4 is to automatically identify and analyze the propagandistic content within memes. Upon processing the meme, the system utilizes GPT-4V to perform a comprehensive analysis that includes the following tasks executed in a single prompt:

1. **Analysis of Meme Text:** The model interprets the text within the meme, considering its semantic and contextual relevance to the image.

2. **Persuasuin Technique Identification:** The model assesses the meme against a predefined list of propaganda techniques, analyzing both the image and text to identify which, if any, techniques are present.

3. **Overall Propagandistic Determination:** The model concludes whether the meme is propagandistic, utilizing both the visual and textual content, and the above analysis, to inform its judgment.

The output of this analysis is structured as a JSON object, which includes three key attributes:



(a) (b)

Figure 2: Label distribution analysis for the multilabel task. (a) **Label frequencies.** The dataset is highly imbalanced with the number of labeled data for each label ranging from as little as 67 samples to as many as 4401 samples. (b) **Label count frequencies**. The majority of samples contain 1 to 4 persuasion techniques. A few samples contain 5 or more techniques and the maximum label count per sample is 8 techniques.

- **Description:** A description of the meme, obtained through multimodal analysis of both the visual and the textual contents of the meme.
- **Techniques:** A list of propaganda techniques identified in the meme.
- **Propagandistic:** A value indicating whether the meme is considered to be propagandistic or not.

This structured output enables a clear, concise, and automated method for identifying and categorizing memes by their propagandistic content, allowing such output to be used in other formulations and experiments that we outline in this paper.

3.2 Early Fusion for Multilabel Persuasion Identification

Our method for multilabel persuasion technique identification in memes incorporates an *early fusion* strategy, utilizing embeddings from both text and image modalities to enrich the feature space. This approach, illustrated in Figure 3, involves two key steps:

1. Embeddings Extraction:
   - We use a multilingual MPNet model (Song et al., 2020; Reimers and Gurevych, 2019) to extract embeddings from the meme's text.
   - The multilingual MPNet is also used to generate embeddings from a meme description that was obtained via a VLM as described earlier in Figure 1.
   - A CLIP-ViT-B-32 multilingual model (Radford et al., 2021; Reimers and Gurevych, 2019) processes the meme image alongside the meme text and the VLM-generated description. This model is used to capture the relationship between visual elements and textual information in memes, producing comprehensive multimodal embeddings.
2. Fusion and Classification:
   - The embeddings from the multilingual MPNet (for both meme text and description) and the CLIP-ViT-B-32 model are fused into a single feature vector. This fusion happens before training the classifier, ensuring the classifier operates on a rich representation of each meme. We use logistic regression for classification.
   - Note that the weights of the embedding models (MPNet and CLIP-ViT-B-32) are frozen during training.

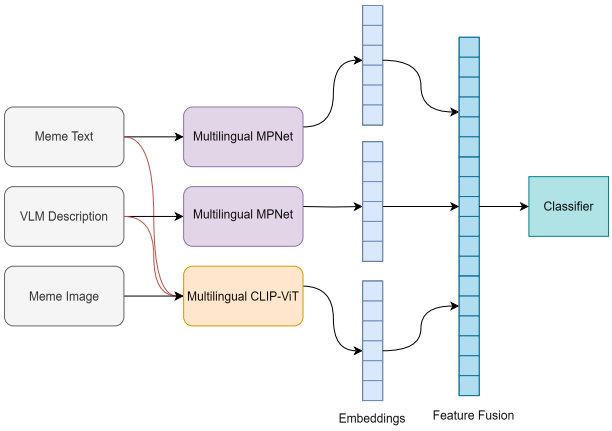


Figure 3: **Early fusion approach overview.** VLM description is obtained as described in Figure 1. CLIP-VIT receives as input all three of the meme image, meme text, and the VLM description. We obtain three separate embeddings which are then concatenated and used to train a classifier.

Team	Avg F1	Macro F1			
		Arabic	English	Bulgarian	North Macedonian
BERTastic (ours)	**66.67**	60.28	71.58	66.21	**68.63**
BCAmirs	65.66	**61.49**	80.34	64.72	56.1
NLPNCHU	63.51	58.52	78.8	64.71	52.0
Snarci	62.52	55.54	79.86	66.78	47.92
LMEME	60.85	36.2	**81.03**	**67.1**	59.08
SheffieldVeraAI	56.16	61.03	64.2	53.62	45.79
BDA	56.1	50.97	79.29	50.62	43.54
DUTIR938	54.52	46.89	80.91	43.41	46.88
HierarchyEverywhere	52.92	56.2	56.31	48.55	50.62
SuteAlbastre	52.05	50.07	80.96	59.45	17.7
Hidetsune	48.95	52.82	71.35	32.67	38.94
IITK	47.71	46.71	48.34	47.26	48.55
nowhash	46.47	49.83	49.84	43.36	42.86
MemeSifters	45.71	55.65	–	61.14	66.03
UMUTeam	19.66	–	78.66	–	–
TUMnlp	19.6	–	78.41	–	–
CodeMeme	19.55	–	78.2	–	–
LomonosovMSU	19.31	–	77.23	–	–
Baseline	18.37	22.7	25.0	16.67	9.09
Scalar	17.54	–	70.15	–	–
WhatsaMeme	12.87	–	51.49	–	–

Table 2: **Results – binary task:** The table shows the results on the test set from the official leaderboard. It shows macro F1 results for all four languages. In addition, we also compute the average macro F1 and sort the teams by this value.

Note that we use multilingual models for all embedding models to ensure our approach generalizes well in the zero-shot scenario. We opted to use a multilingual MPNet which was trained on parallel data on 50+ languages that include Arabic, Bulgarian, and North Macedonian. This decision was made despite the availability of models like XLM-R (Conneau et al., 2020), which, although powerful, were not trained on parallel datasets and thus might not perform as well across languages especially with only English training data. The same rationale applies on why we selected a CLIP-VIT-B-32 model which has a multilingual text encoder that was trained using multilingual knowledge distillation on parallel data.

4 Results

We show the results on binary and multilabel classification tasks in tables 2 and 3, respectively. For binary classification, our system ranks first on North Macedonian propaganda detection task, and achieves competitive results ranking third on Arabic and Bulgarian. Collectively, we achieve the top rank in average F1 across all four languages. As for multilabel classification, our system suffers from low recall but compensates for that by a very high precision performance. Our system achieves the highest precision on Bulgarian, the second highest on Arabic, English, and North Macedonian, and the top precision score on average across all four languages. This means our system is very conservative in that it only makes predictions it is highly confident of.

We also experimented with other models for both the binary and multilabel tasks as shown in tables 4 and 5, respectively. In the binary setting, we train several models on embeddings obtained using a CLIP-ViT-B-32 model, but the zero-shot VLM approach performs better in comparison. For the multilabel task, we try several approaches. We attach a classification head and fine-tune an MPNet model on the meme descriptions that we obtained as described earlier in Figure 1. However, this did not yield a high performance. We also fine-tune DeBERTa-V3-Large (He et al., 2023) and XLM-R-Large on different subsets of available text (i.e., meme text and the VLM descriptions). Out of all combinations, we observe that DeBERTa performed best when it was fine-tuned on VLM descriptions only. This is likely due to the fact the meme texts are often short, incoherent and do not form complete sentences; hence, we deduce they may contaminate the much more coherent descriptions generated by the VLM. Moreover, we also observe that we cannot achieve competitive results using only the text modality.

In all experiments involving transformers, We fine-tune with early stopping with a patience of three epochs, use a batch size of eight, a learning rate of $5e-5$, and we accumulate the gradients for eight steps making our effective batch size 64. We also use a binary-cross entropy loss and set a classification threshold of 0.5.

Team	Avg P	Arabic			English			Bulgarian			North Macedonian		
		F1	P	R	F1	P	R	F1	P	R	F1	P	R
BERTastic (ours)	**77.66**	38.82	61.29	28.41	61.34	81.58	49.14	54.36	**81.16**	40.86	57.33	86.59	42.85
Baseline	76.11	48.65	**65.0**	38.87	44.71	68.78	33.12	50.0	80.43	36.28	55.53	**90.22**	40.1
HierarchyEverywhere	70.91	43.69	50.99	38.21	**74.59**	**86.68**	65.46	46.41	67.08	35.48	35.69	68.9	24.08
BCAmirs	69.75	**52.61**	55.31	50.17	70.5	78.37	64.06	**62.69**	70.28	56.59	**63.68**	75.02	55.32
NLPNCHU	69.73	48.32	59.47	40.70	70.68	78.16	64.5	54.86	70.69	44.83	48.71	70.58	37.18
SuteAlbastre	58.48	51.61	46.94	**57.31**	68.48	71.78	65.47	61.07	65.96	**56.86**	57.55	49.25	**69.22**
IITK	57.65	45.54	45.73	45.35	63.6	76.29	54.54	44.59	54.08	37.93	44.0	54.48	36.9
BDA	49.15	41.64	38.25	45.68	50.39	51.48	49.34	48.34	52.26	44.97	50.14	54.62	46.34
LomonosovMSU	19.79	–	–	–	65.61	79.15	56.02	–	–	–	–	–	–
TUMnlp	19.52	–	–	–	67.72	78.07	59.79	–	–	–	–	–	–
UMUTeam	19.19	–	–	–	69.0	76.76	62.67	–	–	–	–	–	–
Pauk	18.63	–	–	–	67.53	74.5	61.75	–	–	–	–	–	–
CodeMeme	15.16	–	–	–	66.62	60.66	**73.88**	–	–	–	–	–	–
WhatsaMeme	7.84	–	–	–	36.59	31.34	43.96	–	–	–	–	–	–

Table 3: **Results – multilabel task:** The table shows the results on the test set from the official leaderboard. It shows hierarchical F1, precision (P), and recall (R) results for all four languages. In addition, we also compute the average precision and sort the teams by this value.

Method	Training Data	Macro F1
XGBoost	CLIP-ViT embeddings	64.34
LightGBM	CLIP-ViT embeddings	70.1
SVM	CLIP-ViT embeddings	71.91
Zero-shot VLM	Meme image & text	**75.08**

Table 4: **Additional experiments – binary task.** This table reports results on the development set.

Method	Training Data	F1	P	R
MPNet + FFN	VLM descriptions	36.34	57.62	26.55
DeBERTa-V3-Large	Meme text & VLM descriptions	41.34	31.33	60.72
DeBERTa-V3-Large	Meme text	42.15	29.25	**75.4**
DeBERTa-V3-Large	VLM descriptions	42.93	30.18	74.36
XLM-R-Large	VLM descriptions	43.38	31.98	67.42
XGBoost	BLIP embeddings	47.55	79.79	33.87
Zero-shot VLM	Meme image & text	52.56	48.8	56.94
Early Fusion	See Figure 3	**67.84**	**88.93**	54.83

Table 5: **Additional experiments – multilabel task.** This table reports results on the development set. The values reported here are all hierarchical metrics.

5 Conclusion and Future Work

In this study, we introduced a state-of-the-art multilingual propaganda detection in memes using zero-shot learning with VLMs. Our approach uniquely addressed the complexities of multimodal content in memes, merging visual and textual cues in a manner that comprehensively understands and identifies propagandistic content across languages. Achieving an average macro F1 score of 66.7% across all assessed languages, our system demonstrated high performance over existing methods, particularly excelling in North Macedonian memes and showing competitive performance in Bulgarian and Arabic in the binary setting. In addition, our early fusion technique for identifying persuasion techniques in memes within a hierarchical multi-label classification setting outperformed all other approaches with an average hierarchical precision score of 77.66%.

Looking forward, our research opens several directions for further exploration and improvement. First, the exploration of advanced fusion techniques that could more intricately combine the strengths of textual and visual analyses may yield even higher accuracies in propaganda detection. Additionally, the adaptability and performance of our model in detecting subtler forms of propaganda and across a broader spectrum of languages present an exciting challenge, especially considering the highly nuanced contextual nature of meme content and its cultural intricacies.

Acknowledgments

We would like to thank the anonymous reviewers for their time and valuable insights.

References

Harcourt Brace. 1939. *The Fine Art of Propaganda: A Study of Father Coughlin's Speeches*. Institute for Propaganda Analysis and Lee, A.M.C. and Lee, E.B.

Matt Burgers. 2017. Yup, the russian propagandists were blogging lies on medium too | wired uk.

Hadley Cantril. 1938. Propaganda analysis. *The English Journal*, 27(3):217–221.

Dmitry Chernobrov and Emma L Briant. 2020. Competing propagandas: How the united states and russia represent mutual propaganda activities. *Politics*, 42(3):393–409.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Detecting propaganda techniques in memes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6603–6617, Online. Association for Computational Linguistics.

Violet Edwards. 1938. *Group Leader's Guide to Propaganda Analysis: Revised Edition of Experimental Study Materials for Use in Junior and Senior High Schools, in College and University Classes, and in Adult Study Groups*. Institute for propaganda analysis, Incorporated.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Kristina Hristakieva, Stefano Cresci, Giovanni Da San Martino, Mauro Conti, and Preslav Nakov. 2022. The spread of propaganda by coordinated communities on social media. In *14th ACM Web Science Conference 2022*, WebSci '22, page 191–201, New York, NY, USA. Association for Computing Machinery.

Frederick Lauritzen. 1988. Propaganda art in the postage stamps of the third reich. *The Journal of Decorative and Propaganda Arts*, 10:62–79.

H. Lavine, J.A. Wechsler, and Institute for Propaganda Analysis. 1940. *War Propaganda and the United States*. International propaganda and communications. Yale University Press.

Preslav Nakov, Giovanni Da San Martino, and Firoj Alam. 2022. Fact-checking, fake news, propaganda, media bias, and the covid-19 infodemic. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 1632–1634, New York, NY, USA. Association for Computing Machinery.

OpenAI. 2023. [link].

Jennifer Pan, Zijie Shao, and Yiqing Xu. 2020. The effects of television news propaganda: Experimental evidence from china. *Available at SSRN 3579148*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Francis Rawlinson. 2020. "how press propaganda paved the way to brexit", by francis rawlinson - consilium.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Yoonjung Seo, Joshua Berlinger. 2018. South korea stops blasting propaganda as summit looms | cnn.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2023. Vision-language models for vision tasks: A survey.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Appendix

A Prompt Template

The system illustrated in Figure 1 uses the prompt template illustrated in Figure 4. The technique list consists of the names of the twenty-two techniques available in the dataset.

```
You are a helpful meme propaganda analyst designed to output JSON  without any markdown formatting. The
JSON you output must adhere to JSON rules such as case sensitivity for boolean values, and not having
double quotes inside of strings.

Consider the attached meme with the extracted meme text shown below:
{meme_text}

Provide a propaganda/persuasion analysis of the attached meme image alongside the extracted meme text.
Think of propaganda techniques such as:
{technique_list}

Return a JSON that has these attributes:
- description: a generic English description of the meme. Please don't use double quotes.
- techniques: a list of the techniques that are 100% without a doubt present in the meme
- is_propagandistic: True if meme is propagandistic, False otherwise

Only return the JSON, without any explanation. Make sure the JSON is properly formatted. For example,
to quote something inside of a JSON string, use single quotes. Never use double quotes. Also note that
JSON boolean is case sensitive.

{meme_image}
```

Figure 4: **Prompt template:** This is the template used in the binary classification setting illustrated in Figure 1

RKadiyala at SemEval-2024 Task 8: Black-Box Word-Level Text Boundary Detection in Partially Machine Generated Texts

Ram Mohan Rao Kadiyala
University of Maryland , College Park
rkadiyal@terpmail.umd.edu

Abstract

With increasing usage of generative models for text generation and widespread use of machine generated texts in various domains, being able to distinguish between human written and machine generated texts is a significant challenge. While existing models and proprietary systems focus on identifying whether given text is entirely human written or entirely machine generated, only a few systems provide insights at sentence or paragraph level at likelihood of being machine generated at a non reliable accuracy level, working well only for a set of domains and generators. This paper introduces few reliable approaches for the novel task of identifying which part of a given text is machine generated at a word level while comparing results from different approaches and methods. We present a comparison with proprietary systems , performance of our model on unseen domains' and generators' texts. The findings reveal significant improvements in detection accuracy along with comparison on other aspects of detection capabilities. Finally we discuss potential avenues for improvement and implications of our work. The proposed model is also well suited for detecting which parts of a text are machine generated in outputs of Instruct variants of many LLMs.

1 Introduction

With rapid advancements and usage of AI models for text generation , being able to distinguish machine generated texts from human generated texts is gaining importance. While existing models and proprietary systems like GLTR (Gehrmann et al., 2019), ZeroGPT (ZeroGPT), GPTZero (Tian and Cui, 2023), GPTKit (GptKit), Open AI detector , etc.. focus on detecting whether a given text is entirely AI written or entirely human written , there was less advancement in detecting which parts of a given text are AI written in a partially machine generated text. While some of the above mentioned systems provide insights into which parts of the given text are likely AI generated , these are often found to be unreliable and having an accuracy close or worse than random guessing. There is also a rise in usage of AI to spread fake news and misinformation along with using AI models to modify Wikipedia articles (Vice, 2023). Our proposed model focuses on detecting word level text boundary in partially machine generated texts as part of the SemEval shared task : Multi-generator, Multi-domain, and Multilingual Black-Box Machine-Generated Text Detection(Wang et al., 2024b). This paper also discusses implications of findings , comparisons with different models and approaches , comparison with existing proprietary systems with relevant metrics , other findings regarding AI generated texts. The official submission is DeBERTa-CRF , several other models have been tested for comparison. With new, better, and diverse AI models coming into existence, having a model that can make accurate predictions on unseen domains and unseen generator texts can be useful for practical scenarios.

2 Dataset

Set	Count	Sources	Generators
Train	3649	PeerRead	ChatGPT
Dev	505	PeerRead	ChatGPT
Test	11123	PeerRead	LLaMA2
		OUTFOX	LLaMA2
		OUTFOX	GPT-4

Table 1: Dataset sources and split

The dataset used is part of M4GT-bench Dataset(Wang et al., 2024a) consisting of texts each of which are partially human written and partially machine generated sourced from PeerRead reviews and outfox student essays (Koike et al., 2023) all of which are in English. The genera-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 511–519

tors used were GPT-4(OpenAI, 2024) , ChatGPT , LLaMA2 7/13/70B (Touvron et al., 2023). Table 1 shows the source , generator used and data split of the dataset. The generators were given partially human written essays or partially human written reviews along with problem statements and instructions to complete the text. The proportion of human written content in each of the samples ranged from 0 to 50% in the first part while the rest is machine generated in the training data and varying from 0 to 100% in development and test sets. The length of the texts varied between a single sentence to over 20 with median word count of 212 and mean word count of 248.

3 Baseline

The provided baseline uses finetuned Longformer over 10 epochs. The baseline classifies tokens individually as human or machine generated and then maps the tokens to words to identify the text boundary between machine generated and human written texts. The final predictions are the labels of words after whom the text boundary exists. The detection criteria is first change from 0 to 1 or vice versa. We have tried one more approach by considering the change only if consecutive tokens are the same. The baseline model achieved an MAE of 3.53 on the Development set which consists of same source and generator as the training data. The model had an MAE of 21.535 on the test set which consists of unseen domains and generators.

4 Proposed Model

We have built several models out of which DeBERTa-CRF was used as the official submission. We have finetuned DeBERTa(He et al., 2023), SpanBERT(Joshi et al., 2020), Longformer(Beltagy et al., 2020), Longformer-pos (Longfomer trained only on position embeddings), each of them again along with Conditional Random Fields (CRF)(McCallum, 2012) with different text boundary identification logic by training on just the training dataset and after hyperparameter tuning , the predictions have been made on both development and test sets. CRFs have played a vital role in improving the performance of the models due to their architecture being well suited for pattern recognition in sequential data. The primary metric used was Mean Average Error (MAE) between predicted word index of the text boundaries and the actual text boundary word index. However Mean Average Relative Error (MARE) too was used for a better understanding which is the ratio of MAE and text lenght in words. Some of the plots and information couldn't be added due to page limits and are available `here`. [1] along with the `code used`. [2]. a hypothetical example in Figure 1 demonstrates how the model works. The tokens are classified at first and mapped to words. In cases where part of a word is predicted as human and rest as machine (in case of longer words), the word as a whole is classified as machine generated.

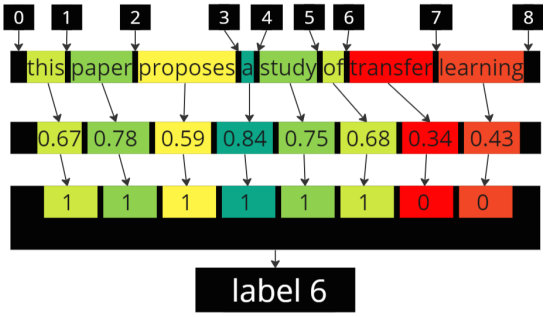


Figure 1: A visual example of working of the model

4.1 Our system

We have used 'deberta-v3-base' along with CRF using Adam(Kingma and Ba, 2017) optimizer over 30 epochs with a learning rate of 2e-5 and a weight decay of 1e-2 to prevent overfitting. other models that have been used are 'Spanbert-base-cased', 'Longformer-base-4096', 'Longformer-base-4096-extra.pos.embd.only' which is similar to Longformer but pretrained to preserve and freeze weights from RoBERTa(Liu et al., 2019) and train on only the position embeddings. The large variants of these have also been tested however the base variants have achieved better performance on both the development and testing datasets. predictions have been made on both the development and testing datasets by training on just the training dataset. Two approaches were used when detecting text boundary 1) looking for changes in token predictions i.e from 1 to 0 or 0 to 1. and 2) looking for change to consecutive tokens i.e 1 to 0,0 or 0 to 1,1. Approach 2 achieved better results than approach 1 in all the cases and was used in the official submission.

[1] more information available at : https://www.rkadiyala.com/papers

[2] Code available at : https://github.com/1024-m/NAACL-2024-SemEval-TASK-8C

4.2 Results

The results from using different models with the two approaches on the development set and the test set can be seen in Table 2. These models have been trained over 30 epochs and the best results were added among the several attempts with varying hyperparameters. The provided baseline however has been trained on just through approach I over 10 epochs using base variant of Longformer. These models have then been used to make predictions on the test set without further training or changes using the set of hyperparameters that produced the best results for each on the development set. However MAE which is the primary metric of the task doesn't take length of the text into consideration, Hence MARE (Mean Average Relative Error) was also calculated for a better understanding.

5 Comparison with proprietary systems

Some of the proprietary systems built for the purpose of detecting machine generated text provide insights into what parts of the text input is likely machine generated at a sentence / paragraph level. Many of the popular systems like GPTZero, GPTkit, etc.. are found to to less reliable for the task of detecting text boundary in partially machine generated texts. Of the existing models only ZeroGPT was found to produce a reliable level of accuracy. For the purpose of accurate comparison percentage accuracy of classifying each sentence as human / machine generated is used as ZeroGPT does detection at a sentence level.

5.1 Results comparison

Since the comparison is being done at a sentence level, In cases where actual boundary lies inside the sentence, calculation of metrics is done on the remaining sentences, and when actual boundary is at the start of a sentence , all sentences were taken into consideration. With regard to predictions, A sentence prediction is deemed correct only when a sentence that is entirely human written is predicted as completely human written and vice versa. The two metrics used were average sentence accuracy which is average of percentage of sentences correctly calculated in each input text, and overall sentence accuracy which is percentage of sentences in the entire dataset accurately classified. The results on the development and test sets are as shown in Table 3. Since its difficult to do the same on 12000 items of the test set , a small section of 500 random samples were used for comparison and were found to perform similar to the development set with a 15-20 percent lower accuracy than the proposed models. Since ZeroGPT's API doesn't cover sentence level predictions , they have been manually calculated over the development set and can be found here. [3].

6 Conclusion

The metrics from Table 3 demonstrate the proposed model's performance on both seen domain and generator data (dev set) along with unseen domain and unseen generator data (test set) , hinting at wider applicability. While there was a drop in accuracy at a word level, there was an increase in sentence level accuracy.

6.1 Strengths and Weaknesses

It was observed that the proprietary systems used for comparison struggled with shorter texts. i.e when the input text has fewer sentences, the predictions were either that the input text is fully human written or fully machine generated leading to comparatively low accuracy.

The average accuracy of sentence level classification for each text length of our model and ZeroGPT can be seen in Figure 2 , Figure 3 respectively. the proposed model overcomes this issue by providing more accurate results even on short text inputs.

The sentence level accuracy did vary considerably while comparing cases where the actual text boundary is at the end of sentence and those where it is mid sentence. The results can be seen in Table 4.

Since the source and generators of texts individually wasn't made available, the comparison between in-domain and out-of-domain texts couldn't be made.

6.2 Possible Improvements

DeBERTa performed better when text boundaries are in the first half of the given text, while Longformer had better performance when the text boundary is in the other half as seen in Figure 4 and Figure 5. In cases where there was a significantly bigger MAE , atleast one of two (DeBERTa and Longformer) had made a very close prediction. There is a possibility that an ensemble of both

[3]ZeroGPT annotations available at : https://docs.google.com/spreadsheets/d/1DOgAZBWQ3G6JtslQwgg9tJiX1WyZt0ajMrr2I9-yfHU/edit?usp=sharing

Dataset →	**Dev set** (Seen Generator)				**Test set** (Unseen Generator)			
Model ↓	**MAE**		**MARE**		**MAE**		**MARE**	
approach →	**I**	**II**	**I**	**II**	**I**	**II**	**I**	**II**
DeBERTa	3.217	3.174	0.0190	0.0185	22.031	19.347	0.1013	0.1006
DeBERTa-CRF	2.311	**2.192**	0.0127	**0.0124**	20.074	**18.538**	0.0919	0.0906
SpanBERT	6.593	5.918	0.0234	0.0221	28.406	25.229	0.1283	0.1274
SpanBERT-CRF	4.855	4.519	0.0196	0.0191	24.283	20.97	0.1216	0.1209
Longformer	3.52	2.878	0.0168	0.0162	25.985	21.177	0.1285	0.1103
Longformer-CRF	2.782	2.41	0.0142	0.0139	20.941	18.943	0.0964	0.0959
Longformer.pos	3.296	3.075	0.0177	0.0174	23.219	19.502	0.1029	0.1022
Longformer.pos-CRF	2.613	2.406	0.0137	0.0135	20.223	18.542	0.0911	**0.0902**
Longformer (baseline)	3.53				21.535			

Table 2: Performance of different models and approaches on dev and test sets

Dev set		
Model	**Accuracy**	**Avg. Acc..**
DeBERTa-CRF	**0.9883**	**0.9848**
Longformer.pos-CRF	0.9806	0.9778
ZeroGPT	0.8086	0.7976
Test set		
Model	**Accuracy**	**Avg. Acc..**
DeBERTa-CRF	**0.9969**	**0.9974**
Longformer.pos-CRF	0.9889	0.9901

Table 3: Performance at sentence level across Development and Test Sets

Model ↓	**mid sent..**	**end of sent..**
DeBERTa-CRF	0.9835	**0.9972**
Longformer.pos-CRF	0.9765	**0.9901**
ZeroGPT	0.7942	**0.8296**

Table 4: Performance of models based on text boundary placement : test set (approach 2)

might perform better, as seen in Table 2, on the test set (unseen generators), while DeBERTa had a better MAE , Longformer had the better MARE. Further, the POS tags of the words pre and post text boundary were examined to find out what led to some cases having higher MAE. Though DeBERTa had better performance, when dealing with very long texts, Longformer might be a better choice. Figure 6 and Figure 7 display the count of data samples in train set and median MAE of those in test set for each POS tags combination pre and post split. The cases where the median MAE was higher (i.e 30 or above) had none or very few samples in the training set. Excluding those cases the new MAE was less than half of what it previously was. Adding more data that covers all cases of pre-split and post-split POS tag words might lead to better results. At a sentence level the accuracy was close to 100 percent excluding the above mentioned samples. Another possible approach worth testing is having a multiplier to the predicted values of each token before classifying as a 0 or 1.

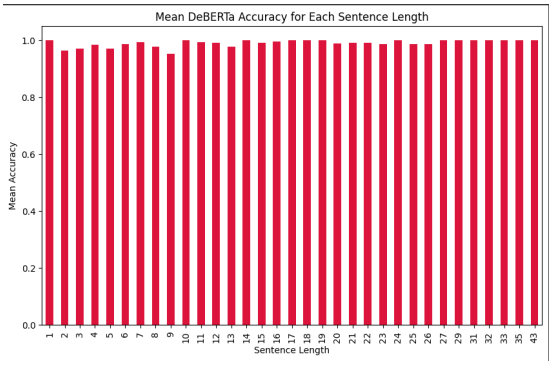


Figure 2: Average sentence accuracy VS number of sentences in test set : DeBERTa-CRF

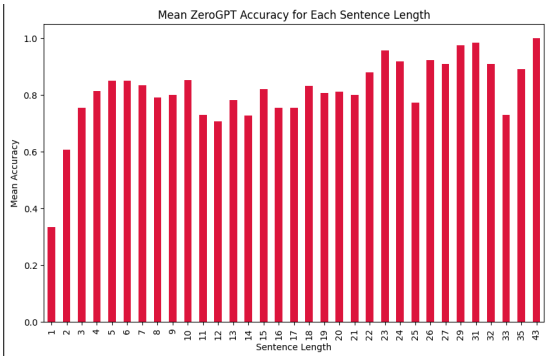


Figure 3: Average sentence accuracy VS number of sentences in test set : ZeroGPT

as seen in Figure 6 and Figure 7, the biggest error cases in pre and post text boundary POS tags were the ones which were not present at all or in

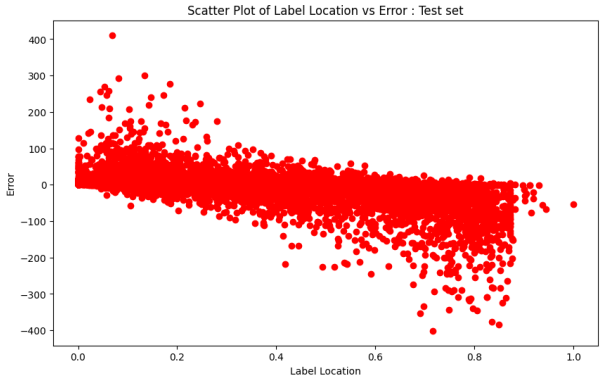


Figure 4: Text boundary location VS MAE in test set : DeBERTa

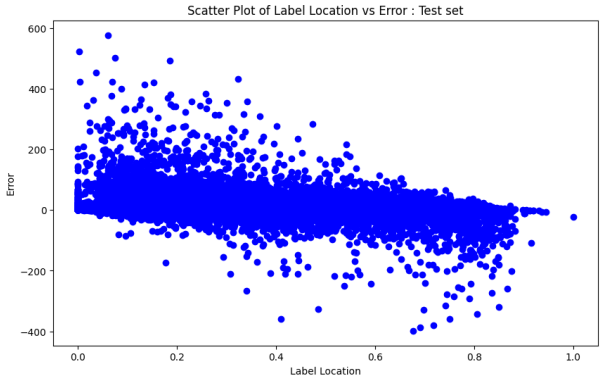


Figure 5: Text boundary location VS MAE in test set : Longformer

very minute amount in the training data, nearly 92 percentage of cases had less than 10 samples to train on and 64 percentage of cases had no samples at all in the training set. A potential solution would be including ample amount of data for all possibilities to cover wider range of texts. This can be done through generating more data by splitting the text at required word boundaries in existing texts and using an LLM to finish the texts.

6.3 Possible Extensions and Applications

The need to detect AI generated content is also prevalent over all languages. While the current model utilizes just English language data, gathering multilingual data and having a multilingual model might also be of great use. With the growth of misinformation and fake news using bots on social media handles(Zellers et al., 2019), being able to detect AI generated texts is of great importance. As most of the texts i.e posts , comments etc.. are shorter in length and difficult to detect, An extension of the current work by training on social media data may yield a good result as demonstrated in Figure 2 and Figure 3. The dataset mostly consisted of texts which are academic related while there is a need to detect machine generated texts in other fields too. Also, It is worth testing the performance of paraphrased data along with the existing data. Since, usage of additional data was prohibited, data augmentation wasn't used in training the current models. It is likely that having more data to cover the cases of pre and post POS tags that weren't present in the training dataset may improve the performance of the models. Some of the other findings are available in Appendix A.

7 Limitations and potential for misuse

While this novel task of detecting text boundaries in partially machine generated texts achieves a high accuracy where one change from human to machine occurs. Being able to handle the cases of multiple changes from human to machine and vice versa is vital. Since having a completely machine generated text and rewriting a few sentences in between or vice versa isn't covered by this work or other existing models, there is a possibility that detection can be evaded this way. There is also a potential for misuse by learning what features and texts caused errors using the proposed models to create texts that can evade detection. The current study covers only two kinds of LLMs i.e GPT and LLaMa. The performance on other types of LLMs is still to be tested. With wide range of available LLMs, training the models over wider range of LLMs might improve performance. The current work focuses on just English texts, however it can be extended to other languages by replacing DeBERTa with mDeBERTa and training on a multilingual corpus. However not all languages are covered by mDeBERTa, this can be a potential issue when dealing with multilingual texts. Another kind of texts that need to be tested upon is where machine generated portions are generated by different generators, and the cases where it is completely machine generated but by different generators. The current corpus used to trained the models is sourced from academic platforms and academic essays, It is necessary to have models to work over a wide variety of texts including cases where it can be in a casual tone. While the current work only considers the first 512 tokens, the longformer version did achieve the same results on unseen generator texts. It is worth looking into how well chunking would work on the deberta model to process longer texts.

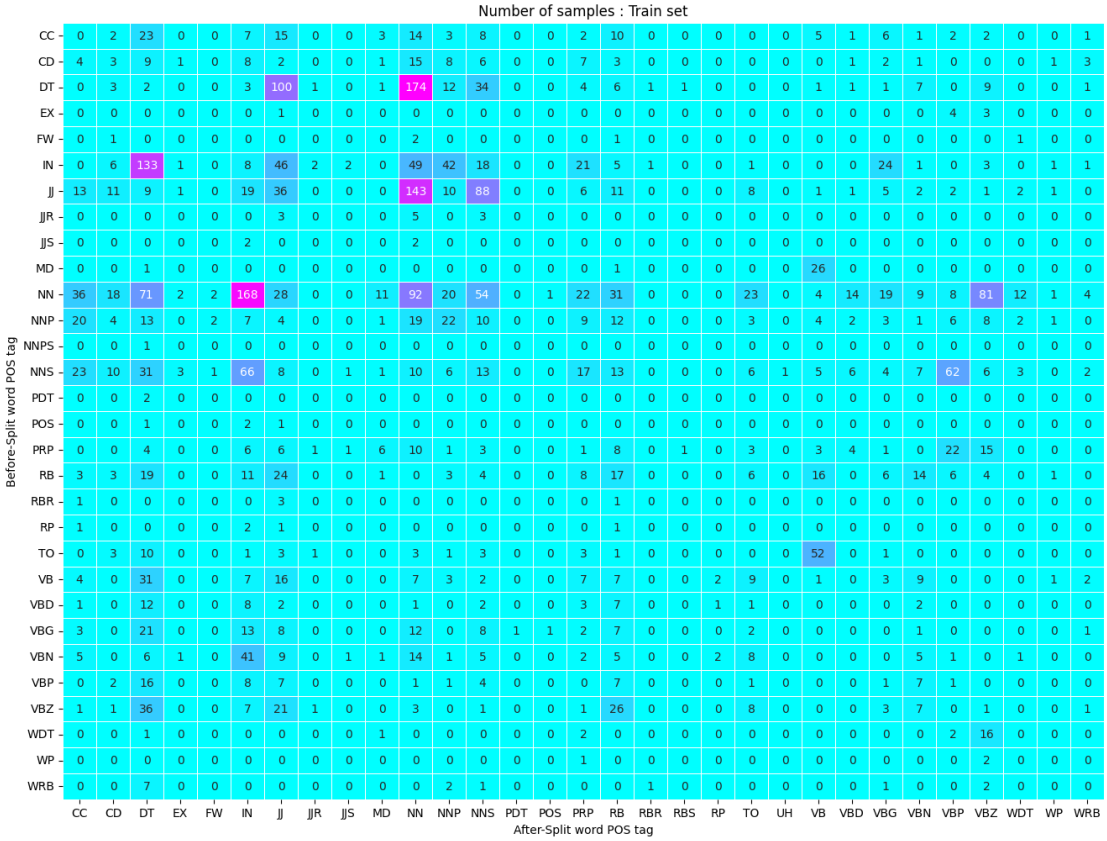


Figure 6: Train set count for each pre and post text boundary POS tag combination

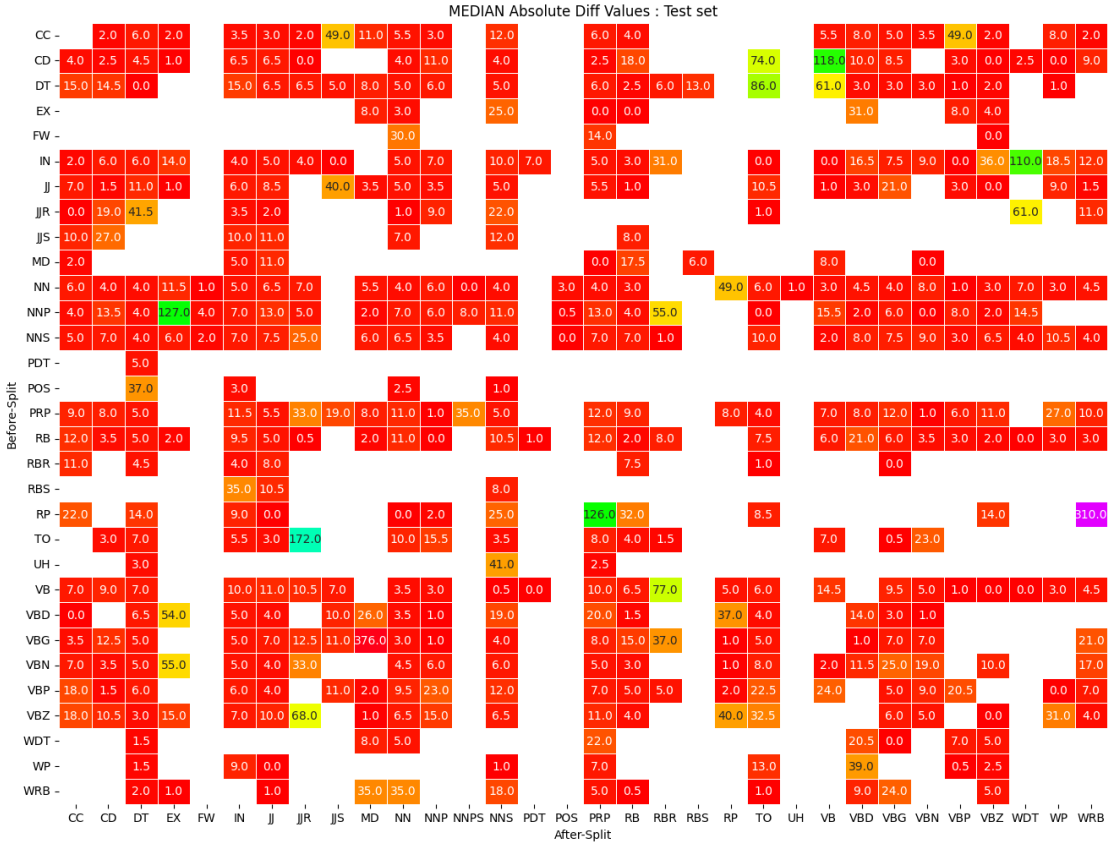


Figure 7: Test set median MAE for each pre and post text boundary POS tag combination

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. *CoRR*, abs/1906.04043.

GptKit. GPTKit: A Toolkit for Detecting AI Generated Text. `https://gptkit.ai/`. Accessed: 2024-02-12.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. In *Transactions of the Association for Computational Linguistics*, volume 8, pages 64–77. MIT Press.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Andrew McCallum. 2012. Efficiently inducing features of conditional random fields.

OpenAI. 2024. Gpt-4 technical report.

Edward Tian and Alexander Cui. 2023. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Vice. 2023. AI Is Tearing Wikipedia Apart. `https://www.vice.com/en/article/v7bdba/ai-is-tearing-wikipedia-apart`. Accessed: 2024-02-12.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

ZeroGPT. Zerogpt : Reliable chat gpt, gpt4 & ai content detector. `https://www.zerogpt.com`.

A Other Plots and information

Some of the information that couldn't be covered due to page limitations along with details for system replication have been added here.

A.1 POS tag usage : human vs machines

It can be seen from Figure 11 , Figure 12 and Figure 10 that machine generated texts had higher share of certain POS tags in the machine generated parts compared to the human written parts. This was observed in all 3 sets, the train and dev had similar distributions as a result of using same generators i.e ChatGPT and the test had a bit of a variation due to multiple different generators i.e LLaMA2 and GPT4. Although the percentile comparison did vary from train, dev and test sets , it was minimal.

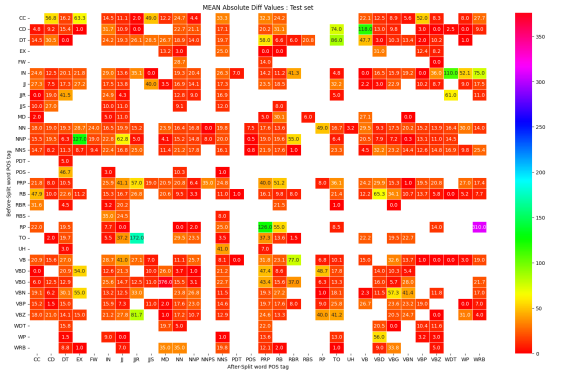


Figure 8: Median MAE based on pre and post text boundary POS tags : DeBERTa-CRF

Figure 9: Median MAE based on pre and post text boundary POS tags : Longformer.pos-CRF

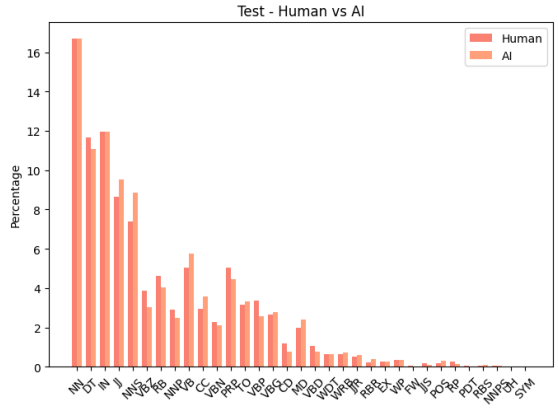


Figure 10: Percentile distribution of each POS tag in test set : human VS machine

A.2 MAE characteristics : DeBERTa vs Longformer

As discussed in the paper , there were some instances where one model performed significantly better than the other as seen in Figure 8 and Figure 9 hinting that an ensemble of both's predictions might yield better results.

B System Description

DeBERTa-CRF was the official submission, longformer.pos-CRF had almost the same performance on the test set. i.e 18.538 and 18.542.

Other models that have been tested but were found to have a big margin of performance with above listed models

Due to time and computational resources limitation, only a part of hyperparameter space was explored.

Official submission model configuration	
Base model	microsoft/deberta--v3-base
Finetuning :	
Learning rate	2×10^{-5}
Weight decay	1×10^{-2}
CRF Dropout rate	75×10^{-4}
Max length	512 tokens
Epochs	30
Optimizer	Adam
Preprocessing	No
Trained on	only train set
Sentence separation	nltk: '!' , '.' , '?'
Hardware	1xV100 GPU 16GB

Table 5: Official submission system description : DeBERTa-CRF

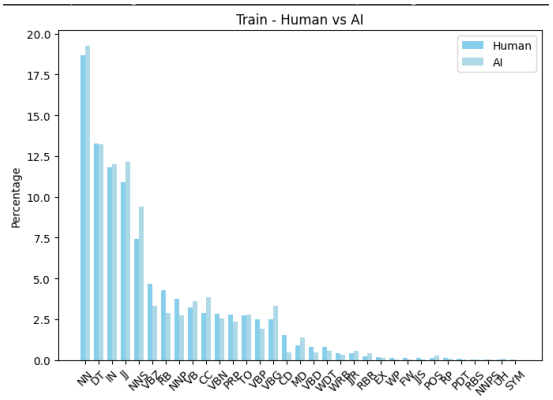


Figure 11: Percentile distribution of each POS tag in train set : human VS machine

Secondary model configuration	
Base model	allenai/longformer-base-4096-extra.pos.embd...
Finetuning :	
Learning rate	2×10^{-5}
Weight decay	1×10^{-2}
CRF Dropout rate	1×10^{-2}
Max length	4096 tokens
Epochs	30
Optimizer	Adam
Preprocessing	No
Trained on	only train set
Sentence separation	nltk: '!' , '.' , '?'
Hardware	1xV100 GPU 16GB

Table 6: Unofficial submission system description : Longformer.pos-CRF

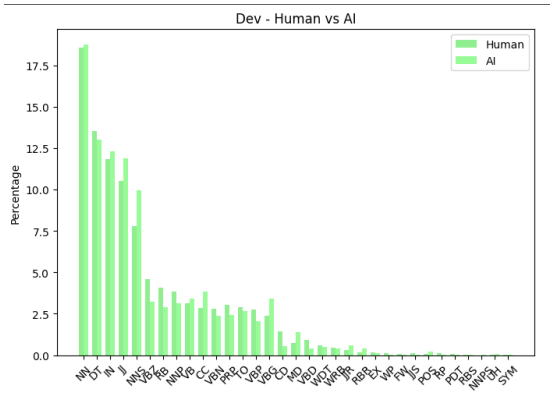


Figure 12: Percentile distribution of each POS tag in dev set : human VS machine

Other models tested
microsoft/deberta-v3-large
microsoft/deberta-v3-small
microsoft/deberta-v3-xsmall
SpanBERT/spanbert-base-cased
SpanBERT/spanbert-large-cased
allenai/longformer-base-4096
allenai/longformer-large-4096
allenai/longformer-large-4096-extra.pos.embd

Table 7: Other models tested as part of the task

C Effect of Text boundary location on performance

The location of text boundaries with respect to length of the text samples are varying over the training and testing set as seen in Figure 13 and Figure 14. Despite training on samples where the text boundaries are in the first half in most of the cases, the models did perform well on the testing set where there is a good amount of samples with text boundaries in later half. This is an area where the proprietary systems struggled.

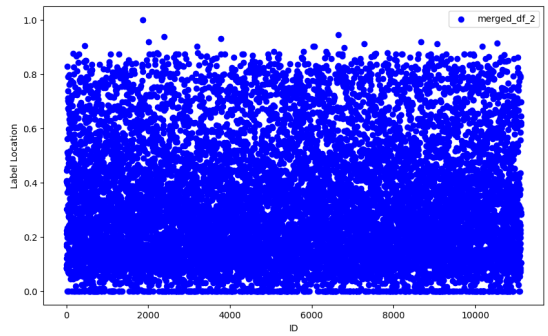


Figure 13: Location of text boundary : testing set

Hyperparameter space explored	
Learning rate	1×10^{-5} 2×10^{-5} 3×10^{-5}
Weight decay	1×10^{-2} 2×10^{-2} 25×10^{-3} 5×10^{-2}
CRF Dropout rates	2×10^{-2} 15×10^{-3} 1×10^{-2} 90×10^{-4} 80×10^{-4} 75×10^{-4} 70×10^{-4} 60×10^{-4}
Max length	512 tokens 4096 *longformer
Epochs	10 to 30
Optimizers	Adafactor Adam
Training data	full train set full train+dev set 80% train set

Table 8: Hyperparameters explored on the models

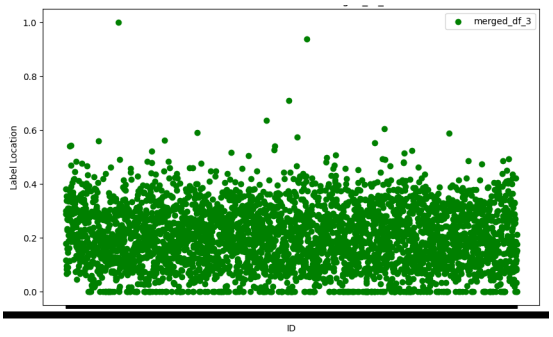


Figure 14: Location of text boundary : training set

TLDR at SemEval-2024 Task 2: T5-generated clinical-Language summaries for DeBERTa Report Analysis

Spandan Das *
Carnegie Mellon University
spandand@andrew.cmu.edu

Vinay Samuel *
Carnegie Mellon University
vsamuel@andrew.cmu.edu

Shahriar Noroozizadeh *
Machine Learning Department
Carnegie Mellon University
snoroozi@cs.cmu.edu

Abstract

This paper introduces novel methodologies for the Natural Language Inference for Clinical Trials (NLI4CT) task. We present TLDR (**T**5-generated clinical-**L**anguage summaries for **D**eBERTa **R**eport Analysis) which incorporates T5-model generated premise summaries for improved entailment and contradiction analysis in clinical NLI tasks. This approach overcomes the challenges posed by small context windows and lengthy premises, leading to a substantial improvement in Macro F1 scores: a 0.184 increase over truncated premises. Our comprehensive experimental evaluation, including detailed error analysis and ablations, confirms the superiority of TLDR in achieving consistency and faithfulness in predictions against semantically altered inputs.

1 Introduction

The Multi-evidence Natural Language Inference for Clinical Trial Data (NLI4CT) task focuses on developing systems that can interpret clinical trial reports (CTRs) and make inferences about them Jullien et al. (2024). The task provides a collection of breast cancer CTRs from ClinicalTrials.gov along with hypothesis statements and labels annotated by clinical experts.

The NLI4CT 2024 task is to classify whether a given CTR entails or contradicts the hypothesis statement. This is challenging as it requires aggregating heterogeneous evidence from different sections of the CTRs like interventions, results, and adverse events. The dataset contains 999 breast cancer CTRs and 2400 annotated hypothesis statements split into training, development, and test sets. The CTRs are summarized into sections aligning with Patient Intervention Comparison Outcome framework. The 2024 SemEval task has the same training dataset as the SemEval 2023 task (Jullien et al., 2023b) but the development and test sets for the 2024 task include interventions of either preserving or inversing the entailment relations for some data points.

In this paper, we introduce TLDR (**T**5-generated clinical-**L**anguage summaries for **D**eBERTa **R**eport Analysis), a novel framework developed for the SemEval Task 2024. Our key contribution is the integration of a T5-based summarization approach to preprocess and condense the premises of clinical reports, which are then analyzed alongside the corresponding statements using DeBERTa, an encoder-only transformer to perform Natural Language Inference (NLI). Our T5-generated summaries address the limitations of small context windows and lengthy premises for this task. This methodology demonstrates a significant improvement in performance on the held-out test set, with our best model achieving an increase of 0.184 in the Macro F1 score compared to using truncated premises and 0.046 over extractive summarized premises. To underscore the efficacy of our approach in this task, we have conducted extensive experiments and ablations, complemented by a thorough error analysis. We also demonstrate the efficacy of our model's performance with regards to consistency and faithfulness against semantically altered inputs. These efforts collectively highlight the robustness and effectiveness of the TLDR framework in addressing the complexities and nuances of NLI task within the domain of clinical report trials.

2 Background

The NLI4CT task requires developing systems capable of NLI from clinical trial reports. In the following section, we examine not only contributions within the SemEval NLI4CT task (Jullien et al., 2023b,a) but also wider advancements in the field.

Transformer Architectures Pretrained transformer models form the backbone of many top-performing systems in the NLI4CT task, with their

*All authors contributed equally to this work.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 520–529

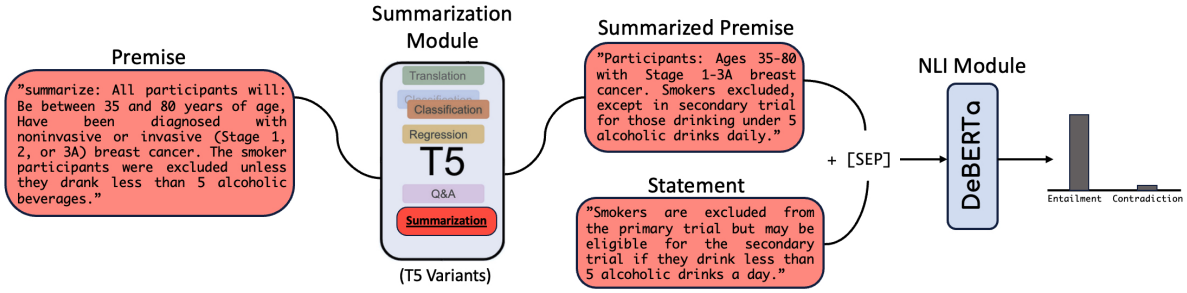


Figure 1: TLDR Model: TLDR processes the clinical report by initially summarizing it using the summarization module (variants of the T5 Model). This summary is then merged with the statement and fed into the fine-tuned DeBERTa model for Natural Language Inference.

architectures being a crucial aspect of their design.

Generative Transformers Generative transformers, which include models like the instruction-tuned Flan-T5 (Kanakarajan and Sankarasubbu, 2023), are particularly adept at generating text and can produce probabilities or direct entailment labels. These models excel in tasks that require the generation of coherent language constructs and have been pretrained on biomedical data, equipping them with the necessary domain knowledge. The work of Zhao et al. (2023) further emphasizes this, showcasing the efficacy of ChatGPT, a generative model, in a multi-strategy system for clinical trial inference, particularly through prompt learning techniques.

Discriminative Transformers Discriminative transformers are employed for classification tasks and include variants of BERT such as BioBERT and ClinicalBERT (Lee et al., 2020; Chen et al., 2023; Vladika and Matthes, 2023). These models have been fine-tuned on domain-specific data to enhance their understanding of medical texts. DeBERTa architecture (He et al., 2020), which improves upon BERT with disentangled attention and enhanced masking, is also included in this category. These approaches were successfully incorporated by Chen et al. (2023) and Alameldin and Williamson (2023) utilizing transformer-based models for both evidence retrieval and entailment determination in NLI4CT task of 2023.

3 System overview

In this section, we explain our methodologies to address the complexities of the Natural Language Inference for Clinical Trials (NLI4CT) task. We primarily focus on the utilization of large language models such as T5 (Raffel et al., 2020) and BERT-based architectures (Devlin et al., 2018), to implement innovative techniques to manage the challenge of the long premise lengths and token size limitations. As mentioned in Section 2, DeBERTa is a top-performing model for NLI tasks. However, it has a limitation in clinical NLI such as the task in NLI4CT due to its restricted context length, making it difficult to include both the premise and the statement. We therefore propose to fine-tune T5 to summarize long clinical premises, which can then allow us to leverage DeBERTa's discriminative power for clinical NLI with extended premises. We call our model TLDR for **T**5-generated clinical-**L**anguage summaries for **D**eBERTa **R**eport Analysis. Our full pipeline can be seen in Figure 1.

In this section, we delve into the fine-tuning of these models for our clinical NLI task and employ specialized summarization strategies using variants of the T5 model, balancing between zero-shot and fine-tuned approaches. *

3.1 TLDR: NLI with DeBERTa enhanced by T5 Generated Summaries

We experimented with DeBERTa, an encoder-only transformer model and its ability for natural language inference for the NLI4CT task. Taking the context length into account, the full premise does not fit into our model so we would have to get a shortened version of the premise. To achieve this, we generate a summary of each premise using a T5 variant that can fit the context length of DeBERTa (see Section 3.2). This shortened premise along with the full statement of each data point in the dataset is then used to make a prediction of entailment or contradiction for that statement. Our input to the DeBERTa model has the following form:

*Our code is available at: https://github.com/Shahriarnz14/TLDR-T5-generated-clinical-Language-for-DeBERTa-Report-Analysis

[CLS] + shortened premise + [SEP] + statement

where the [CLS] token is used for the binary classification of entailment or contradiction.

3.2 Summarization Techniques for Premises

The need for summarizing the premises arises due to the long length of the premises and the limited input length of several of the top performing BERT-based architectures such as DeBERTa that is utilized in this paper.

In our tasks, there are two types of data points: "Single" and "Comparison". For "Single" data points, the statement is checked for entailment with only a primary premise. In "Comparison" instances the statement is checked for entailment when compared to a primary as well as a secondary premise. For each of the summarization methods below, we consider summarizing both the primary and second premises independently of each other as well as summarizing the primary and secondary premises combined together.

3.2.1 Inference Only

Encoder-Decoder architectures such as the T5 model have shown strong capabilities for summarizing text. To this end we used several T5 variants to produce a summary for each premise with a maximum source length of 2048 tokens and a maximum generated summary of 300 tokens thereby decreasing the size of the premise to a size that would enable our full input to be passed into the DeBERTa model.

3.2.2 Fine-tuning

Our objective for fine-tuning T5-based models is to enable the generation of summaries closely mirroring ground truth statements for entailed premise-statement pairs. We fine-tune T5 variants on a dataset exclusively comprising pairs labeled "Entailment". "Contradiction" instances are excluded to avoid confusion of the model in generating summaries as they include contradictory information about the premise. We prepend the premises with "summarize:", as inputs to T5 and treat the corresponding ground truth statements as labels, aiming to align the generated summaries with these statements. This approach ensures the model is trained to produce summaries that effectively encapsulate the entailed information.

4 Experimental setup

In this section, we provide a comprehensive outline of our experimental methods, setting the stage for a detailed discussion on each technique employed.

The fine-tuning of each module of our model was done on the training set and evaluated on the development set. Upon selecting the best performing model from the development set, we then evaluated our model on the held-out test. The performance reported in Section 5 is on this held-out test set.

4.1 The Discriminative NLI Module of TLDR

For the shortened premise and the statement pairs, we fine-tuned an NLI fine-tuned version of the base DeBERTa-v3 model (Tran et al., 2023) from Hugging Face for Entailment and Contradiction binary classification. Specifically we used the `cross-encoder/nli-deberta-v3-base`. This model was trained using SentenceTransformers Cross-Encoder class on the SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) datasets and provided improved NLI performance over standard DeBERTa-v3-base model. The tokenizer used was also taken from Hugging Face and was the tokenizer corresponding to the DeBERTa model that we used. The fine-tuning of our DeBERTa model was done on the training set for 40 epochs at a learning rate of 5×10^{-5} and evaluated on the development set to pick the best performing model for the final evaluations.

Importantly, DeBERTa has a maximum input size of 512 tokens. Therefore, the combined length of the shortened premise and statement pair is required to fit into this size (the average statement length in the training data was 110 tokens). The description of the different premise shortening techniques we employed is outlined below.

4.2 The Summarization Module of TLDR

We employed different variants of T5 to generate summaries of the clinical premises as a way to shorten them. In this section, we explain each of these approaches.

Zero-Shot T5 We utilized the Hugging Face API to summarize premises with *google/flan-t5-base*, limiting summaries to 300 tokens. For instances containing both primary and secondary premises, we conducted two different approaches: (1) separate summaries for each and (2) a combined summary for both premises together. The first approach

involved generating individual summaries for primary and secondary separately, and creating a single summary for a concatenated version of both premises in the second approach.

Fine-tuned T5 For Summary Generation To fine-tune T5 for summary generation, we filtered our training and development datasets to only include instances labeled as entailment, using these as ground truth for summarized premises. In the case of entailment, we claim that since the statement is an accurate representation of the premise it also serves as an appropriate ground truth label for a summarized premise. We opted for the *t5-small* model due to resource constraints, acknowledging this may impact comparison fairness with zero-shot models. Our fine-tuning utilized the ROUGE-1 metric which compares unigrams between the predicted and the label summary. We fine-tuned for $2, 5, 7,$ and 10 epochs at a learning rate of 2×10^{-5}, weight decay of 0.01 and a batch size of 4.After fine-tuning, we generated separate summaries for primary and secondary premises using this model.

Fine-tuned SciFive We followed the exact same procedure explained above from the fine-tuned T5 for the *razent/SciFive-base-Pubmed_PMC* model from Hugging Face. For this T5 variant, we aimed leverage the fact that the SciFive model was trained on biomedical literature (Phan et al., 2021) which is similar in domain to our setting. Here, we used the same hyperparameters for fine-tuning T5. After fine-tuning was complete, for data instances with two premises, we generated the summaries by separately summarizing the primary and secondary premises and then combining the two.

4.3 Summarization Ablations

We used two ablation strategies for TLDR's summarization module. Instead of using a T5 variant for summarizing each premise, these two ablations included: (1) naively truncating premises and (2) using a traditional extractive summarization.

Truncated Premises To fit within the 512 token limit of DeBERTa, we tokenized the combined statement and premises, subtracting the statement's token count and an additional 10 tokens from 512 to stay under the limit. More concretely, for $x = 512 -$ (# of statement tokens) $- 10$, in single-premise data points, we truncated the primary premise to x tokens and in comparison data points with both a primary and a secondary premise, we truncated each premise to $\frac{x}{2}$ tokens.

Extractive Summarization We also used an extractive summarization technique with TF-IDF for our ablation experiments, which evaluates word significance in a document against a corpus. To avoid data leakage and maintain evaluation accuracy, the TF-IDF vectorizer was applied exclusively to the training dataset. It was then used to summarize texts within the training, development, and test sets. Summaries were produced by identifying and selecting sentences with the highest TF-IDF scores, adhering to a 300-word limit for conciseness. This extractive summarization ablation, based on TF-IDF, allows us to compare summarization techniques and their impact on DeBERTa's NLI model performance, emphasizing the importance of feature representation and data handling.

4.4 Performance Metrics

We report the held-out test set performance metrics of the different variants of TLDR model. Specifically, we are report Macro F1, Precision, and Recall as measures of the prediction performance. In addition, we also report Faithfulness and Consistency.

Faithfulness assesses if a system predicts accurately for the right reasons, gauged by its response to **semantic altering** interventions. With N contrast set statements x_i, original y_i, and predictions $f()$, faithfulness is calculated using Equation 1.

$$Faithfulness = \frac{1}{N} \sum_{1}^{N} |f(y_i) - f(x_i)|$$
$$x_i \in C : \text{Label}(x_i) \neq \text{Label}(y_i), \text{ and } f(y_i) = \text{Label}(y_i) \quad (1)$$

Consistency evaluates a system's output uniformity for semantically equivalent inputs, focusing on identical predictions under **semantic preserving** interventions. This ensures semantic representation consistency, regardless of prediction accuracy. For N statements x_i in contrast set (C), with original y_i and predictions $f()$, consistency is determined using Equation 2.

$$Consistency = \frac{1}{N} \sum_{1}^{N} 1 - |f(y_i) - f(x_i)| \quad (2)$$
$$x_i \in C : \text{Label}(x_i) = \text{Label}(y_i)$$

5 Results

The performance of our TLDR variants and ablations on the NLI task for clinical report trials is summarized in Table 1. Our results showcase

Table 1: Performance of TLDR Variants and Ablations on the Held-Out Test Set

Method	Macro F1	Precision	Recall	Faithfulness	Consistency
Ablations: DeBERTa Methods					
DeBERTa + Truncated Premise(s)	0.474	0.432	0.524	0.573	0.542
DeBERTa + Extractive Summarized Premise(s)	0.612	0.584	**0.643**	**0.615**	**0.590**
TLDR Methods					
TLDR (flan-T5-base - Zero-Shot Combined Premises)	0.599	0.628	0.573	0.409	0.540
TLDR (flan-T5-base - Zero-Shot Distinct Premises)	0.633	0.624	0.642	0.502	0.574
TLDR (T5-small - Best fine-tuned)	0.635	0.676	0.599	0.436	0.557
TLDR (SciFive-base - Best fine-tuned)	**0.658**	**0.684**	0.633	0.501	0.581

the effectiveness of different approaches, with notable variations in Macro F1, Precision, Recall, and Faithfulness, and Consistency across methods.

The TLDR methods showed the most promising results. The best-performing model based on prediction performance was the TLDR method using SciFive-base for fine-tuned summarization of distinct premises, achieving the highest Macro F1 score of 0.658 and the best precision of 0.684. This approach also demonstrated a strong recall at 0.633 thereby indicating a strong balance between precision and recall. A close second was the TLDR approach with fine-tuned T5-small, which attained a Macro F1 score of 0.635 and precision of 0.676 and recall of 0.599, indicating the efficacy of fine-tuning on task-specific data. In the appendix, Table 2 illustrates the impact of varying the total number of fine-tuning epochs for T5-small and SciFive on the downstream NLI task. The main finding is that unlike T5-small that longer fine-tuning degraded the downstream performance, SciFive required longer fine-tuning steps to see improvements in TLDR's downstream prediction. We suspect this is due to the fact that the original fine-tuning of SciFive had degraded its summarization performance compared to T5-small and thus it required to be fine-tuned for longer.

For the ablations methods, we observed an improvement in performance when using extractive summarization instead of naively truncating the input. The method utilizing truncated premises achieved a Macro F1 score of 0.474, with the lowest precision of 0.432 among all methods and the lowest recall of 0.524 among all methods, suggesting this strategy as being inappropriate for dealing with the context length issue for this particular task. The extractive summarization approach yielded a much higher Macro F1 score of 0.612 with a much higher precision at 0.584 and recall at 0.643 thereby clearly outperforming the truncated premises strategy. Note that these are ablations of our introduced more complete TLDR model. Extractive summarization proved the best strategy for the faithfulness and consistency metrics with a faithfulness score of 0.615 and a consistency of 0.590. This result likely stems from the fact that extractive summarization of premises maintains key tokens from the original premises, which preserves the core semantics. This can then result in modifications from the contrastive set's interventions in the test data to be more straightforwardly mapped and identified by the TLDR's NLI module.

For a more detailed error analysis and exploration of each model's performance across various sections of the clinical trial, refer to Appendix B. The main takeaway is that TLDR methods leveraging fine-tuning and distinct premise summarization, consistently outperformed simpler input models across all clinical trial sections, demonstrating the significance of specialized summarization and training techniques in managing the challenges of lengthy premises in clinical trials.

6 Conclusion

In this paper, we introduced TLDR (**T**5-generated clinical-**L**anguage summaries for **D**eBERTa **R**eport Analysis) tailored for clinical NLI tasks, with a focus on NLI4CT 2024. Our investigation reveals that strategies incorporating SciFive for distinct premise summarization and fine-tuning for summarization to better align with entailed statements about the premises markedly improve handling of clinical language and reasoning complexities. Despite the challenges posed by lengthy premises in clinical reports, our TLDR methods, particularly those employing advanced summarization through fine-tuning, consistently demonstrated superior performance over simpler methods. This underscores the importance of model adaptation and the strategic selection of summarization techniques in enhancing the accuracy and reliability of NLI tasks

within the clinical domain.

Looking ahead, a promising avenue for future work involves the use of the best encoder-decoder transformer summarization model for each specific section of clinical reports. This approach could potentially improve the overall performance of NLI where we saw fine-tuned SciFive was particularly better in some sections of the clinical report and T5-based summaries were better at some other sections. Continued exploration and refinement of these models are essential to further advance the field of NLI in clinical applications.

7 Acknowledgements

The authors would also thank Dr. Daniel Fried, Saujas Vaduguru, and Kundan Krishna for helpful discussions.

S.N. was supported by Natural Sciences and Engineering Research Council of Canada .

References

Ahamed Alameldin and Ashton Williamson. 2023. Clemson nlp at semeval-2023 task 7: Applying gatortron to multi-evidence clinical nli. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1598–1602.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Chao-Yi Chen, Kao-Yuan Tien, Yuan-Hao Cheng, and Lung-Hao Lee. 2023. NCUEE-NLP at SemEval-2023 task 7: Ensemble biomedical LinkBERT transformers in multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 776–781, Toronto, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama AI research at SemEval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003, Toronto, Canada. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Cong Dao Tran, Nhut Huy Pham, Anh Tuan Nguyen, Truong Son Hy, and Tu Vu. 2023. ViDeBERTa: A powerful pre-trained language model for Vietnamese. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1071–1078, Dubrovnik, Croatia. Association for Computational Linguistics.

Juraj Vladika and Florian Matthes. 2023. Sebis at SemEval-2023 task 7: A joint system for natural language inference and evidence retrieval from clinical trial reports. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1863–1870, Toronto, Canada. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Xiaofeng Zhao, Min Zhang, Miaomiao Ma, Chang Su, Yilun Liu, Minghan Wang, Xiaosong Qiao, Jiaxin Guo, Yinglu Li, and Wenbing Ma. 2023. Hw-tsc at semeval-2023 task 7: Exploring the natural language inference capabilities of chatgpt and pre-trained language model for clinical trial. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1603–1608.

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. THiFLY research at SemEval-2023 task 7: A multi-granularity system for CTR-based textual entailment and evidence retrieval. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1681–1690, Toronto, Canada. Association for Computational Linguistics.

A Fine-Tuning Encoder-Decoder Transformers for Specific Summary Generation

In this section, we present how the NLI performance varies across different fine-tuning steps of the T5 variants.

In the fine-tuning results of Table 2, we observe a peak performance at two epochs for both T5-small and SciFive-base models, with a slight performance drop as the number of epochs increases. The T5-small model shows a trade-off between precision and recall, reaching its highest recall at five epochs but with better overall performance at two epochs. The SciFive-base model maintains a consistent precision after two epochs, but the recall fluctuates, suggesting that the optimal number of training epochs is crucial for maintaining model balance and preventing overfitting. We believe the reason that SciFive required longer fine-tuning epochs compared to T5-small is because the initial fine-tuning of SciFive diminished its summarization capabilities relative to T5-small, necessitating extended fine-tuning to restore performance.

Table 2: Fine-Tuning Results

Method	Macro F1	Precision	Recall	Faithfulness	Consistency
T5 Fine-Tuning					
TLDR (T5-small - 2 Epochs)	0.605	0.580	0.633	0.557	0.593
TLDR (T5-small - 5 Epochs)	0.635	0.676	0.599	0.436	0.557
TLDR (T5-small - 7 Epochs)	0.601	0.564	0.644	0.597	0.590
TLDR (T5-small - 10 Epochs)	0.603	0.580	0.628	0.608	0.587
SciFive Fine-Tuning					
TLDR (SciFive-base - 2 Epochs)	0.570	0.528	0.620	0.552	0.580
TLDR (SciFive-base - 5 Epochs)	0.628	0.652	0.606	0.470	0.563
TLDR (SciFive-base - 7 Epochs)	0.613	0.588	0.639	0.560	0.589
TLDR (SciFive-base - 10 Epochs)	0.658	0.684	0.633	0.501	0.581

B Error Analysis

In this discussion, we first delve into a detailed error analysis in subsection B.1, examining model performances across different sections and types of clinical trial data. The results presented is on the practice held-out test set. Following this, in subsection B.4 we explore the prediction agreement among the various TLDR models.

B.1 Error Analysis

The first part of our error analysis focuses on the performance of TLDR methods and ablation DeBERTa models across different sections of clinical trial reports: Eligibility, Adverse Events, Results, and Interventions. The analysis is grounded in Macro F1 scores, average premise lengths, and average statement lengths, as detailed in the Tables 4,5,6,7.

Eligibility Section In the Eligibility section, the TLDR method using flan-T5-base for zero-shot distinct premises achieved the highest Macro F1 score (0.626), indicating its effectiveness in summarizing and understanding eligibility criteria. Notably, this model and the best fine-tuned SciFive-base model managed to significantly reduce the average premise length while maintaining high performance, suggesting that effective summarization can aid in dealing with long premises typically found in this section.

Adverse Events Section The Adverse Events section saw the highest Macro F1 score (0.775) with the TLDR method using T5-small, best fine-tuned. This model also had one of the shortest average premise lengths, indicating that fine-tuning on specific data, even with shorter premise lengths, can yield high accuracy in identifying adverse events. The DeBERTa models performed relatively well in this section but were outperformed by the TLDR approaches.

Results Section For the Results section, the TLDR method with fine-tuned SciFive-base, showed the best performance with a Macro F1 of 0.67. Interestingly, this model also had the shortest average premise length, suggesting a strong correlation between effective summarization and model performance. The low performance of the SciFive-base zero-shot model indicates that domain-specific fine-tuning is crucial for understanding complex results data that we gain from generating summaries similar to the entailment statements. Note that similar to (Zhou et al., 2023) where they observed SciFive showed superior performance for results with numerical data, we also see the gain in using SciFive when used for the results section.

Intervention Section In the Intervention section, the TLDR flan-T5-base method for zero-shot combined premises showed the highest Macro F1 score (0.647). This suggests that the model's ability to synthesize information from combined premises is particularly effective in understanding intervention-related data.

In the second part of our error analysis, distinct trends are revealed in the performance of the DeBERTa and TLDR models when dealing with single and comparison premises in clinical trial reports. This distinction is crucial as single premises present a straightforward context, whereas comparison premises involve juxtaposing and interpreting two separate contexts. These results are presented in the appendix in Table 3.

Single Premises In the single premise category, the TLDR methods generally outperform their ablated counterparts using DeBERTa models. The TLDR method using flan-T5-base for zero-shot distinct premises and the best fine-tuned SciFive-base model both achieved a Macro F1 score of 0.642, the highest in this category. This indicates their robustness in handling singular, focused clinical contexts. Notably, these models significantly reduced the average premise length, with the best fine-tuned SciFive-base model achieving the shortest length, which suggests an effective summarization capability that preserves essential information that we achieve by attempting to generate summaries that are aligned with the entailment statements.

Comparison Premises For comparison premises, where the task involves analyzing and relating two different contexts, the TLDR models still outperform the DeBERTa models, but with a slight decrease in overall effectiveness compared to single premises. The highest Macro F1 score is 0.631 with the TLDR flan-T5-base for zero-shot distinct premises. The best fine-tuned models, both T5-small and SciFive-base, also show strong performance in this more complex scenario. Interestingly, the average premise lengths are longer for comparison premises across all models, underscoring the increased complexity and information content in these types of premises.

Across both single and comparison premises, TLDR methods demonstrate superior performance, especially in handling and effectively summarizing complex clinical data. The shorter average premise lengths in the best-performing models suggest that their summarization strategies are successful in distilling essential information without losing context crucial for NLI tasks. This is particularly evident in the comparison premises, where managing two contexts simultaneously is a challenging task. In conclusion, the type of premise (single or comparison) significantly impacts the model's performance, with TLDR methods showing robustness in both scenarios. The findings emphasize the importance of tailored summarization techniques and model fine-tuning to handle the varying complexities in clinical trial reports.

B.2 Results Split By Type

Below in Table 3 we include results that were obtained split on whether the data instance was a single instance meaning only a primary premise was given or a comparison instance where both a primary and a secondary instance was given.

B.3 Results Split By Section Type

Below in Tables 4, 5 6, 7 we include results that were obtained split on Eligibility, Adverse Events, Results, and Intervention sections respectively. These are the different sections in the Clinical Trial Reports that the statement in the data instance is referring to

B.4 Prediction agreement across various model

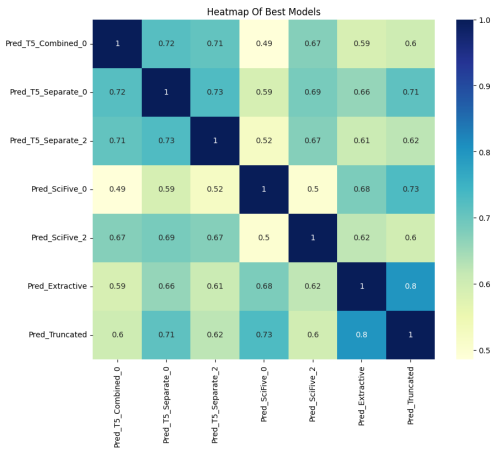


Figure 2: Heatmap comparing the model predictions

In our comparative analysis of model predictions, depicted in Figure 2, we observe distinct patterns of agreement among the various models tested. Notably, T5-based models exhibit a high degree of consistency in their predictions, as evidenced by the darker blue 3x3 square in the top left corner of the heatmap. This suggests a strong underlying

Table 3: Type Results

Method	Macro F1	Avg Premise Len	Avg Premise - Ent	Avt Premise - Con	Avg Statement Len
Single					
Ablations: DeBERTa Methods					
DeBERTa + Truncated Premise(s)	0.484	1152.2	1149.5	1154.9	121.2
DeBERTa + Extractive Summarized Premise(s)	0.574	725.0	724.1	725.8	121.2
TLDR Methods					
TLDR (flan-T5-base - Zero-Shot Combined Premises)	0.642	334.0	333.3	334.8	121.2
TLDR (flan-T5-base - Zero-Shot Distinct Premises)	0.644	276.2	275.8	276.6	121.2
TLDR (SciFive-base - Zero-Shot Premises)	0.427	542.6	545.3	539.9	121.2
TLDR (T5-small - Best fine-tuned)	0.637	196.0	196.2	195.9	121.2
TLDR (SciFive-base - Best fine-tuned)	0.642	79.2	79.1	79.3	121.2
Comparison					
DeBERTa Methods					
DeBERTa + Truncated Premise(s)	0.52	2270.8	2273.9	2267.8	145.7
DeBERTa + Extractive Summarized Premise(s)	0.522	956.5	958.2	954.9	145.7
TLDR Methods					
TLDR (flan-T5-base - Zero-Shot Combined Premises)	0.587	407.5	406.4	408.7	145.7
TLDR (flan-T5-base - Zero-Shot Distinct Premises)	0.631	448.9	449.1	448.8	145.7
TLDR (SciFive-base - Zero-Shot Premises)	0.471	1046.1	1043.5	1048.8	145.7
TLDR (T5-small - Best fine-tuned)	0.626	371.7	371.5	372.0	145.7
TLDR (SciFive-base - Best fine-tuned)	0.612	150.5	150.2	150.7	145.7

Table 4: Section Results - Eligibility

Eligibility					
Method	**Macro F1**	**Avg Premise Len**	**Avg Premise - Ent**	**Avt Premise - Con**	**Avg Statement Len**
Ablations: DeBERTa Methods					
DeBERTa + Truncated Premise(s)	0.395	3776.0	3776.0	3776.0	147.4
DeBERTa + Extractive Summarized Premise(s)	0.444	1517.7	1517.7	1517.7	147.4
TLDR Methods					
TLDR (flan-T5-base - Zero-Shot Combined Premises)	0.574	636.6	636.6	636.6	147.4
TLDR (flan-T5-base - Zero-Shot Distinct Premises)	0.626	418.8	418.8	418.8	147.4
TLDR (SciFive-base - Zero-Shot Premises)	0.448	1070.3	1070.3	1070.3	147.4
TLDR (T5-small - Best fine-tuned)	0.537	383.0	383.0	383.0	147.4
TLDR (SciFive-base - Best fine-tuned)	0.613	137.3	137.3	137.3	147.4

Table 5: Section Results - Adverse Events

Adverse Events					
Method	**Macro F1**	**Avg Premise Len**	**Avg Premise - Ent**	**Avt Premise - Con**	**Avg Statement Len**
Ablations: DeBERTa Methods					
DeBERTa + Truncated Premise(s)	0.583	496.1	496.1	496.1	109.9
DeBERTa + Extractive Summarized Premise(s)	0.646	496.6	496.6	496.6	109.9
TLDR Methods					
TLDR (flan-T5-base - Zero-Shot Combined Premises)	0.641	243.0	243.0	243.0	109.9
TLDR (flan-T5-base - Zero-Shot Distinct Premises)	0.699	292.4	292.4	292.4	109.9
TLDR (SciFive-base - Zero-Shot Premises)	0.43	678.1	678.1	678.1	109.9
TLDR (T5-small - Best fine-tuned)	0.775	217.2	217.2	217.2	109.9
TLDR (SciFive-base - Best fine-tuned)	0.675	107.0	107.0	107.0	109.9

Table 6: Section Results - Results

Results					
Method	**Macro F1**	**Avg Premise Len**	**Avg Premise - Ent**	**Avt Premise - Con**	**Avg Statement Len**
Ablations: DeBERTa Methods					
DeBERTa + Truncated Premise(s)	0.45	2022.4	2013.9	2030.8	145.5
DeBERTa + Extractive Summarized Premise(s)	0.518	971.4	971.6	971.1	145.5
TLDR Methods					
TLDR (flan-T5-base - Zero-Shot Combined Premises)	0.575	358.0	352.6	363.3	145.5
TLDR (flan-T5-base - Zero-Shot Distinct Premises)	0.622	437.7	435.6	439.8	145.5
TLDR (SciFive-base - Zero-Shot Premises)	0.333	1040.8	1035.0	1046.6	145.5
TLDR (T5-small - Best fine-tuned)	0.604	320.2	318.2	322.2	145.5
TLDR (SciFive-base - Best fine-tuned)	0.67	132.0	130.5	133.5	145.5

Table 7: Section Results - Intervention

Intervention					
Method	**Macro F1**	**Avg Premise Len**	**Avg Premise - Ent**	**Avt Premise - Con**	**Avg Statement Len**
Ablations: DeBERTa Methods					
DeBERTa + Truncated Premise(s)	0.558	752.9	752.9	752.9	135.1
DeBERTa + Extractive Summarized Premise(s)	0.543	439.1	439.1	439.1	135.1
TLDR Methods					
TLDR (flan-T5-base - Zero-Shot Combined Premises)	0.647	252.1	252.1	252.1	135.1
TLDR (flan-T5-base - Zero-Shot Distinct Premises)	0.602	339.0	339.0	339.0	135.1
TLDR (SciFive-base - Zero-Shot Premises)	0.516	526.7	526.7	526.7	135.1
TLDR (T5-small - Best fine-tuned)	0.615	246.9	246.9	246.9	135.1
TLDR (SciFive-base - Best fine-tuned)	0.555	98.3	98.3	98.3	135.1

similarity in how these models process and interpret the summaries for the test data. In contrast, the SciFive-based models display a marked divergence in their prediction patterns. The fine-tuned version of the SciFive model, in particular, demonstrates a significant shift in its predictions, aligning with the positive performance changes highlighted in previous sections. Furthermore, the two ablated versions employing either truncated premises or extractive summarization exhibit a high level of agreement in their predictions, as indicated by the dark blue 2x2 square in the heatmap's bottom right corner. This consistency points to the robustness of these ablation methods in maintaining prediction alignment. Overall, these findings underscore the varying degrees of prediction agreement across different model architectures and highlight the impact of model-specific features and training approaches on prediction outcomes in clinical NLI tasks.

ignore at SemEval-2024 Task 5: A Legal Classification Model with Summary Generation and Contrastive Learning

Binjie Sun
School of Information Science and Engineering
Yunnan University
sunbinjie@stu.ynu.edu.cn

Xiaobing Zhou
School of Information Science and Engineering
Yunnan University
zhouxb@ynu.edu.cn

Abstract

This paper describes our work for SemEval-2024 Task 5: The Legal Argument Reasoning Task in Civil Procedure. After analyzing the task requirements and the training dataset, we used data augmentation, adopted the large model GPT for summary generation, and added supervised contrastive learning to the basic BERT model. Our system achieved an F1 score of 0.551, ranking 14th in the competition leaderboard. Our system achieves an F1 score improvement of 0.1241 over the official baseline model.

1 Introduction

In Task 5 of SemEval-2024: The Legal Argument Reasoning Task in Civil Procedure (Bongard et al., 2022), we expect to reason about legal arguments in civil actions, as shown in Table 1. The dataset for this task comes from a textbook for law students, and we believe it is a complex task that can be benchmarked against modern legal language models. Task 5 proposes a novel NLP task from the US civil procedure domain that is beneficial to the quest to improve modern legal language models.

The foundation model we choose is Legal-BERT (Chalkidis et al., 2020), which collects different English LEGAL texts from multiple domains (e.g., legislation, court cases, contracts) for pre-training. Compared with other models such as LEGAL-RoBERTa (Chalkidis* et al., 2023), it can handle this task data better. Based on that, a great variety of strategies have been tested along with our exploration, such as summary generation, data augmentation (DA), and contrastive learning.

Data analysis for this task revealed that the dataset size was relatively small (only 666 entries), yet each data point contains substantial information. In such a language environment, we realize using and enriching data fully is very important. We used generative summarization, contrastive learning, and data augmentation to train the model, which led to our system ranking 14th in this task.

2 Related Work

Legal information is mostly expressed in the form of text, such as legal cases, bills, contracts, legislation, and so on. Therefore, legal text processing is an important area of NLP, including classifying legal topics (Nallapati and Manning, 2008), generating rulings based on what the court has already done (Ye et al., 2018), etc. In the past, some traditional machine learning methods like SVM bag of words (Aletras et al., 2016; Medvedeva et al., 2018) performed worse than neural models on legal tasks. The use of generic pre-trained models becomes the new paradigm, such as Legall Longformer (Chalkidis* et al., 2023) and Italian-Legal-BERT (Licari and Comandè, 2022). Data augmentation is a mature method for expanding a dataset when there is little training data, and in this case, we are not using external data but rather taking full advantage of the various fields of the provided data.

Task 5 is a small sample task, and we adopt contrastive learning to distinguish them from different samples by grouping similar samples together, hoping to learn from the intrinsic structure of the data. We use triples as a loss function(Schroff et al., 2015), and according to the characteristics of our task, we use a supervised contrast learning(Khosla et al., 2020) algorithm, where the triples are anchor points, positive samples, and negative samples.

3 System Overview

Our baseline system simply feeds Legal-BERT with two pieces of text, classifies its output [CLS] tokens, and scores their similarity with the human-annotated data by cross-entropy loss training. All the optimized strategies discussed below are based on this framework, and the overall framework of our final system is shown in Figure 1. After train-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 530–535

key	value
Introduction	My students always get confused about the relationship between removal to federal court and personal jurisdiction. Suppose that a defendant is sued in Arizona and believes that she is not subject to personal jurisdiction there .[...]Fed. R. Civ. P. 4(k)(1)(A). I've stumped a multitude of students on this point. Consider the following two cases to clarify the point.
Question	7. A switch in time. Yasuda, from Oregon, sues Boyle, from Idaho, on a state law unfair competition claim, seeking $250,000 in damages. He sues in state court in Oregon.[...] Five days after removing, Boyle answers the complaint, including in her answer an objection to personal jurisdiction. Boyle's objection to personal jurisdiction is
Answer Candidate	not waived by removal, but will be denied because the federal courts have power to exercise broader personal jurisdiction than the state courts.
Label	0

Table 1: An example in the training set.

ing with all positive policies, we ensemble the best model on each fold for the final prediction.

3.1 Data Augmentation

In this task, we augment the training data in two ways. First, we combine the explanation, the question, and the complete analysis corresponding to the answer to form new positive sample data by utilizing the fields of the complete parsing of the answer. Second, the analysis corresponding to the wrong answer is combined with the answers to other questions to form new negative sample data.

In the original training dataset, the data ratio of positive and negative samples is 505:161 (505 samples have a label of 0). Through the above data augmentation methods, the data is expanded and the data set is balanced.

3.2 Summary Generation

The task requires giving a question and possibly correct answers to determine whether the answer is correct or incorrect. We should also consider short introductions to the question topic rather than directly using the question and answer fields of the sample data. For legal texts, the same question will have different answers in different contexts, and the differences in the answers are often huge.

We plan to concatenate the explanation and the question together to form the text for the first input system and the answer as the text for the second input. We choose Legal-Bert to handle up to 512 tokens, while most of the training data have more than 512 tokens, and the distribution of sample lengths in the training data set is shown in figure 2. We tried different truncation methods (direct truncation, sliding window truncation) to improve the performance of the model and finally found that using GPT3.5 to generate a summary of the context can achieve a better result than truncation processing.

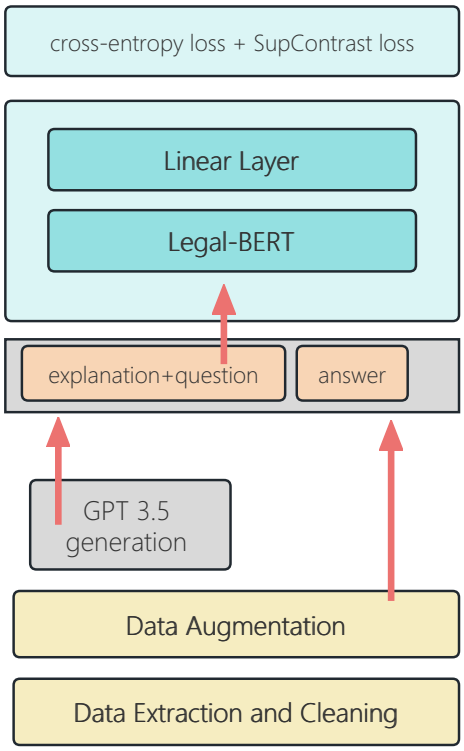


Figure 1: The overall framework of our system proposed for SemEval-2024 Task 5.

The specific treatment we adopt in direct and sliding window truncation is as follows. In direct truncation, we used the explanation and the question field 'l' space. Then, after the mosaics of the

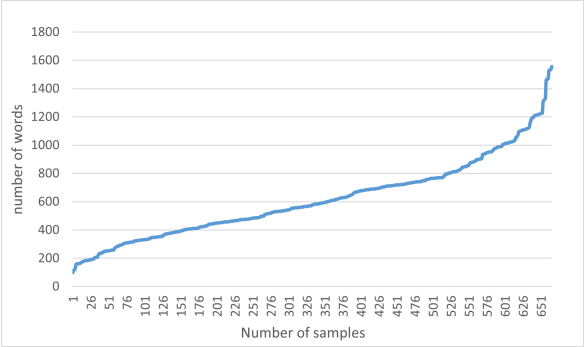


Figure 2: Sample length in the training dataset

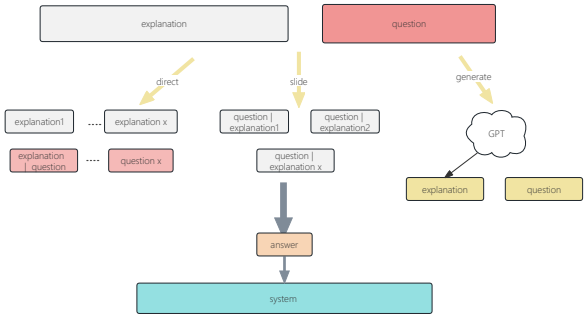


Figure 3: Explanation processing method

strings, separated by spaces counting more than 150 words, as a new sample data, the "id complete" field is used to identify the segmentation. In the sliding window, the basic strategy is the same as the above. Still, in each segmentation, the question's existence is guaranteed, and the part of the 150 words minus the question is explained on the concatenation. The specific process is shown in the figure 3.

However, we found their shortcomings in the above two processing methods. Directly truncating the simple truncated data will lead to the information in the question field with some sample data, either only the context or only the original question information. In sliding window truncation, although the original question field is preserved, we believe that the key information of the explanation is not uniformly distributed in the sentence. Therefore, we adopt GPT3.5 to generate the corresponding summary explanation according to the question pair context.

We believe that the important information to be extracted from the introduction usually involves key sentences, general sentences, and important details, which will affect whether the candidate's answer to the question is correct or not. Abstract generation for introducing a problem uses large models' good generalization ability to extract and compress this general knowledge. This can preserve the integrity of the information and capture the information from a broader perspective than the segmentation method.

3.3 Supervised Contrastive Learning

Contrastive learning aims to learn a data representation by maximizing the similarity between relevant samples and minimizing the similarity between irrelevant samples. In order to better fit this classification task, we use the Supervised Contrastive Learning strategy (Khosla et al., 2020; Chen et al., 2020), in which points belonging to the same class are pulled together in their own space. In contrast, points belonging to different classes are separated.

In a batch input, we treat the samples containing the original answer field as anchors, the newly added complete analysis of the answer as positive samples, and the remaining samples under the same question as negative samples. The contrastive loss under this triplet is shown in Eq 1, which is:

$$L_A = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(f(x_i),f(x_i^+))}}{e^{s(f(x_i),f(x_i^+))}+\sum_{j=1}^{N}e^{s(f(x_i),f(x_j^-))}}$$

where x_i is the input anchor, x_i^+ is the positive sample, x_i^- is the negative sample, $s(f(x_i), f(x_i^+))$ is the similarity measure function, and the inner product is commonly used.

We want to evaluate from an overall point of view, so we combine the cross-entropy loss and contrastive loss as the loss function of the model to train, and the loss function of the model is shown in Eq 2:

$$L = \frac{1}{2}\cdot(L_{CE}(y_i,\hat{y}_i) + L_A(x_i, x_i^+, x_i^-))$$

Where y_i is the ground truth, $\hat{y}_i$ is the predicted value, and x_i, x_i^+, x_i^- are the anchors, positive samples, and negative samples in the upper segment. Each data in the dataset has y_i and $\hat{y}_i$ after training, but only one kind of sample corresponds to the contrastive loss.

4 Experimental setup

4.1 Dataset Description

The training and validation sets contain 666 and 84 samples, respectively. Each sample contains a question, answer, label, analysis(excerpt from complete analysis relevant to answer candidate), complete

analysis(Glannon's explanation for the solution of the question), and explanation(topical introduction, additional context for question, potentially empty) fields. The test set contains 98 examples and has only question, answer, and explanation fields.

The task purpose is, given a question with a likely correct answer and a short introduction to the question topic, to determine whether the answer candidate is correct or incorrect. Each of these sample data does not exist independently, and most of them are 4 to 6 samples in the same group. This means that the questions and contexts of these four data are consistent, and the answers and analyses are different. Specific examples are shown in Table 2.

The following are the specific available fields and what they represent for the samples in the dataset:

- <question> 6. Any port in a storm. Cullen, a Vermont citizen, has an accident with Barnabas, a citizen of California, and Tecumseh, a New Yorker, in California. She sues Barnabas and Tecumseh for negligence in state court in Albany, New York, alleging negligence. She serves Barnabas with process in the ...
- <answer> a motion to transfer the case to a California court under 28 U.S.C. §1404(a).
- <analysis> A isn't right either. Section 1404(a) is a federal statute, authorizing a federal court to transfer a case to another federal court. It does not govern the state courts. There is no transfer statute allowing state courts in one state to transfer cases to ...
- <complete analysis> This question provides a nice little recap of various jurisdiction and venue issues. Barnabas wants out of the New York state court. What motion is likely to do the trick? Removal seems like an option, though of course he'd still have to litigate in New York. Remember that you can only ...
- <explanation> So, venue is the "third ring" in choosing a proper court, along with personal jurisdiction and subject matter jurisdiction. If all three rings are satisfied, the court has the power to hear the case. However, it doesn't always do so. Sometimes a case is filed in a court that has subject matter jurisdiction over the case, personal jurisdiction over the defendant, and is a proper venue under ...

Column	Train	Dev	Test
idx	true	true	true
question	true	true	true
answer	true	true	true
label	true	true	false
analysis	true	true	false
complete analysis	true	true	false
explanation	true	true	true

Table 2: Components of the dataset.

- <label> False

4.2 Dataset Split

We split the processed training set and validation data set into 10 subsets without intersection and randomly split them into units of the same background-size, which ensures that each set has the same proportion of positive and negative samples as the original full set. Ten-fold cross-validation is used, and the results are shown as averages to ensure that the strategy used is maximally effective on the final test set.

4.3 Pre-processing

The legal data in all datasets were provided to us by email by the task organizers. After getting the original file in CSV format, we remove the file headers and re-add the file headers based on data splitting or summarization. After the initial processing of the data, we split the data into a mini-batch of 8 according to the needs of contrastive learning, where the first data is the anchor, the second data is the positive example, and the third to six data are the negative examples. In the cleaning process, we mainly remove some dirty format data, such as some missing field data.

4.4 Evaluation Metrics

Task 5 has two evaluation metrics which are F1 score and precision, The F1 score is common in evaluating binary classification tasks, especially when the classes are imbalanced, it is more representative than precision or recall. The F1 score can range from 0 to 1, with values closer to 1 indicating better performance.

4.5 Others

Hyperparameter tuning was not a critical point of our work. Still, we tested several values over a small range as they did influence our decisions

System	F1 score
pratice augmentation	
Baseline	42.69
+ DA	46.96
evaluation augmentation	
Baseline	42.96
+ DA	50.33
+ Summary Generation	53.59
+ SupContrast Learning	**55.10**

Table 3: Best results with training methods we used.

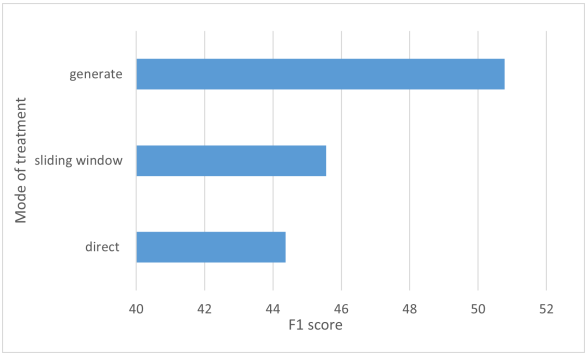


Figure 4: Summary generation effect comparison

about how well the policy worked (see Appendix). In addition, to help the reader replicate our experiments, details of tools and libraries are provided (see Appendix).

5 Results

5.1 Overall Performance

Finally, according to the official scoring system, our system got 0.551 on the test set and ranked 14th. As results are shown in Table 3, all the strategies presented in Section 3 produced positive effects, and we discuss the effects of these strategies one by one in the following subsections. For convenience, all the results from our experiments are multiplied by 100.

5.2 Data Augmentation

To verify whether the augmented dataset plays a positive role, we train with the augmented dataset in the Practice phase of the competition, which provides the official baseline, and this is the gap between the two baselines in Table 3.

As you can see from the top of Table 3, there is a significant increase, which is consistent with our inference that a richer training set is beneficial to build a more accurate system, and the way we augment the data is to some extent a multi-perspective supplement to the original data (from the analysis of the problem).

5.3 Summary Generation

As introduced in Section 3.2, we are aware of the importance of the corresponding explanation of the question. We propose several different ways to include segmentation fields and generate summaries. However, we are not sure which method is effective in collecting the characteristics of the data. Therefore, we tried each method, and the results are shown in Figure 4. Compared with direct truncation, the sliding window truncation method has an improvement of about 1 point, and the generated summary can be improved by about 4 points on this basis.

Obviously, through comparison, it is found that compared with direct truncation and sliding window truncation, the context summary generated by using a large model can better represent the features of the data. By comparing the direct truncation method and the sliding window truncation method, it can also be seen that the effect of the sliding window is better than the direct truncation to a certain extent, which conforms to our basic cognition that explanation is crucial in problem reasoning. Whether a candidate answer to a question is correct or not depends on the context of the question, that is, the relevant introduction.

5.4 Supervised Contrastive Learning

As mentioned in Section 3.3, contrastive learning is incorporated into our system. The loss function of our system is composed of a combination of cross-entropy loss and contrastive loss. We show the output of the cross-entropy loss and contrastive loss in some epochs of training and find that the contrastive learning function values are larger than the cross-entropy loss, and their magnitude is usually about double.

Through our final experimental results, as shown in the table, we can find that after the addition of contrastive learning, our system can learn more general features by reducing the distance between positive examples and away from the distance between negative examples, which increases the adversarial robustness of the model.

6 Conclusion

By deploying various optimization methods, including data augmentation, summary generation,

and supervised contrastive learning, we build a conceivably powerful system to reason about the task of legal argumentation in civil litigation. And ranked 14th in the evaluation stage competition with a 0.551 F1 score in the officially organized competition.

In future work, one is that law is a serious domain, and we plan to guide the model by prior knowledge. We also plan to incorporate domain-specific knowledge into the exercises and analyses of the law school textbooks under study. Second, we consider whether we can better model long texts by using tools external to the model to assist in processing long texts and optimizing the model.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: a natural language processing perspective. *PeerJ Comput. Sci.*, 2:e93.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The Legal Argument Reasoning Task in Civil Procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, UAE. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Ilias Chalkidis*, Nicolas Garneau*, Catalina Goanta, Daniel Martin Katz, and Anders Søgaard. 2023. LeX-Files and LegalLAMA: Facilitating English Multinational Legal Language Model Development. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Daniele Licari and Giovanni Comandè. 2022. ITALIAN-LEGAL-BERT: A Pre-trained Transformer Language Model for Italian Law. In *Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management*, volume 3256 of *CEUR Workshop Proceedings*, Bozen-Bolzano, Italy. CEUR. ISSN: 1613-0073.

Maria Medvedeva, Michel Vols, and Martijn Wieling. 2018. Judicial decisions of the european court of human rights: looking into the crystall ball. In *Proceedings of the Conference on Empirical Legal Studies in Europe 2018*.

Ramesh Nallapati and Christopher D. Manning. 2008. Legal docket classification: Where machine learning stumbles. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 438–446, Honolulu, Hawaii. Association for Computational Linguistics.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823. IEEE Computer Society.

Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. 2018. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1854–1864, New Orleans, Louisiana. Association for Computational Linguistics.

A Appendix

Table 4 and Table 5 provide the details of the corresponding hyperparameters and libraries.

Hyperparameter	Range/Value
Epoch	30 - 50
Batch Size	8
Warm-up-nums	10
Learning Rate	3e-5∼5e-5

Table 4: Main hyper-parameters tuned in our system.

Tools & Libraries	Version
NumPy	1.22.3
pandas	1.4.0
Python	3.7.10
PyTorch	1.13.0
Transformers	4.15.0

Table 5: Main tools and libraries used in our system.

Samsung Research China-Beijing at SemEval-2024 Task 3: A multi-stage framework for Emotion-Cause Pair Extraction in Conversations

Shen Zhang[⋆], **Haojie Zhang**[⋆✉], **Jing Zhang**
Xudong Zhang, Yimeng Zhuang, Jinting Wu
Samsung R&D Institute China-Beijing
{shen02.zhang, tayee.chang, jing97.zhang,
xudong.z1, ym.zhuang, jinting01.wu}@samsung.com

Abstract

In human-computer interaction, it is crucial for agents to respond to human by understanding their emotions. Unraveling the causes of emotions is more challenging. A new task named Multimodal Emotion-Cause Pair Extraction in Conversations is responsible for recognizing emotion and identifying causal expressions. In this study, we propose a multi-stage framework to generate emotion and extract the emotion causal pairs given the target emotion. In the first stage, Llama-2-based InstructERC is utilized to extract the emotion category of each utterance in a conversation. After emotion recognition, a two-stream attention model is employed to extract the emotion causal pairs given the target emotion for subtask 2 while MuTEC is employed to extract causal span for subtask 1. Our approach achieved first place for both of the two subtasks in the competition.

1 Introduction

Comprehending emotions plays a vital role in developing artificial intelligence with human-like capabilities, as emotions are inherent to humans and exert a substantial impact on our thinking, choices, and social engagements (Wang et al., 2023b). Dialogues, being a fundamental mode of human communication, abound with a variety of emotions (C. et al., 2008; Poria et al., 2019; Zahiri and Choi, 2017; Li et al., 2017; Xia and Ding, 2019; Ding et al., 2020; Wei et al., 2020; Fan et al., 2020). Going beyond simple emotion identification, unraveling the underlying catalysts of these emotions within conversations represents a more complex and less-explored challenge (Wang et al., 2023b). Hence, (Wang et al., 2023a, 2024) introduces a novel undertaking known as Recognizing Emotion Cause in Emotion-Cause-in-Friends (ECF). ECF contains 1,344 conversations and 13,509 utterances where 9,272 emotion-cause pairs are annotated, covering textual, visual, and acoustic modalities. All utterances are annotated by one of the seven emotion labels, which are neutral, surprise, fear, sadness, joy, disgust, and anger. Within ECF, a significant task is identified as Emotion-Cause Pair Extraction in Conversations (ECPEC). ECPEC is responsible for identifying causal expressions related to a specific utterance in conversations where the emotion is implicitly expressed. ECPEC provides two Multimodal Emotion Cause Analysis in Conversations (ECAC) subtasks:

- Subtask 1: Textual Emotion-Cause Pair Extraction in Conversations. Given a conversation containing the speaker and the text of each utterance $U = [U_1, U_2, ...U_n]$, the model is aim to predict emotion-cause pairs, which include emotion utterance's emotion category and the textual cause span in a specific cause utterance (e.g. U3_joy, U2_"You made up!").
- Subtask 2: Multimodal Emotion Cause Analysis in Conversations. Given a conversation including the speaker, text and audio-visual clip for each utterance, the model is aim to predict emotion-cause pairs, which include emotion category and a cause utterance (e.g. U5_Disgust, U5).

To address the above problem, Wang et al. (2023a) proposed a two-step approach. First, they extract the emotional utterances and causal utterances by a multi-task learning framework and then pair and filter them. Zhao et al. (2023) proposes an end-to-end method by leveraging multi-task learning in a pipeline manner. However, these methods still suffer from low evaluation performances.

Motivated by the phenomenon that the performance of the emotion recognition of utterances in

⋆: equal contributions. ✉: Corresponding Author.
Shen Zhang is in charge of the basic subtask-emotion recognition in conversation (ERC) and Haojie Zhang is responsible for the pipeline framework and causal pair extraction and causal span extraction subtasks.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 536–546

a conversation harnessed by the traditional manner is generally poor, we design a new pipeline framework. Firstly we utilize the Llama-2-based InstructERC (Lei et al., 2023a) to extract the emotion category of each utterance in a conversation. Then we consider the emotion causal pair extraction as the causal emotion entailment subtask and employ a two-stream attention model to extract the emotion causal pairs given the target emotion. For the causal span extraction, we employ MuTEC (Bhat and Modi, 2023) which is an end-to-end multi-task learning framework.

2 Related Works

2.1 Emotion Recognition in Conversation

Emotion recognition in conversation (ERC), which is a task to predict emotions of utterances during conversations, is crucial in both of the two ECAC subtasks. The existing methods can be divided into graph-based, RNN-based, Transformer-based, LLM-based, and knowledge-injecting methods.

Graph-based methods (Shen et al., 2021b; Li et al., 2024; Zhang et al., 2019; Taichi et al., 2020; Ghosal et al., 2019) aims to represent the correlations between emotions of utterances and speakers in the conversations. RNN-based methods (Hu et al., 2023; Lei et al., 2023c; Majumder et al., 2019; Hazarika et al., 2018; Poria et al., 2017) using GRU and LSTM (Wang et al., 2020) to capture the dependency of interlocutors and emotions of utterances. To model the emotional states during long-range context, Transformer-based methods (Song et al., 2022; Liu et al., 2023b; Chudasama et al., 2022; Shen et al., 2021a; Hu et al., 2022) utilize encoder-decoder framework or encoder-only models, such as BERT (Li et al., 2020) and RoBERTa (Kim and Vossen, 2021), to establish the correlation between long-range emotional states during conversations. Considering more than seven utterances in single conversation input, InstructERC (Lei et al., 2023b) defines the ERC task as a generative task based on LLMs, which unifies emotion labels between three common ERC datasets and utilizes auxiliary tasks (speaker identification and emotion prediction) by using instruction template to capture speaker relationships and emotional states in future utterances. Knowledge-injecting methods (Freudenthaler et al., 2022; Ghosal et al., 2020; Zhong et al., 2019; Zhu et al., 2021; Lei et al., 2023b) use external knowledge to analyze conversation scenarios.

2.2 Emotion Causes in Conversations

Poria et al. (2021) introduces the task of recognizing emotion causes in conversations and introduce two novel sub-tasks: Causal Span Extraction (CSE) and Causal Emotion Entailment (CEE), designed to identify the emotion cause at the span-level and utterance-level, respectively.

Causal Emotion Entailment Poria et al. (2021) define CEE as a classification task for utterance pairs and establish robust Transformer-based baselines for it. Wang et al. (2023a) introduces a multimodality conversation dataset Emotion-Cause-in-Friends (ECF) and propose a two-step approach to extract the causal pairs. They first extract the emotion utterances and the potential causal utterances individually and then pair and filter them. Li et al. (2022) introduce the social commonsense knowledge to propagate causal clues between utterances. Zhao et al. (2023) propose the Knowledge-Bridged Causal Interaction Network (KBCIN), which integrates commonsense knowledge (CSK) as three bridges called semantics-level bridge, emotion-level bridge and action-level bridge.

Causal Span Extraction involves identifying the causal span (emotion cause) for a given non-neutral utterance. Poria et al. (2021) first introduces the subtask and employs the pre-trained Transformer-based model to formulate the Causal Span Extraction as the Machine Reading Comprehension (MRC). Bhat and Modi (2023) propose a multi-task learning framework to extract the causal pairs and causal span in an utterance in a joint end-to-end manner. Besides, they also propose a two-step approach consisting of Emotion Prediction (EP), followed by Causal Span (CSE).

3 System Overview

3.1 System Architecture

The overview of the architecture of our proposed model is shown in Figure 1. The InstructERC aims to extract the emotion of utterances. TSAM model is a two-stream attention model utilized to extract the causal pairs given the predicted emotion utterance. The MuTEC is an end-to-end network designed to extract the causal span based on the causal pair extraction.

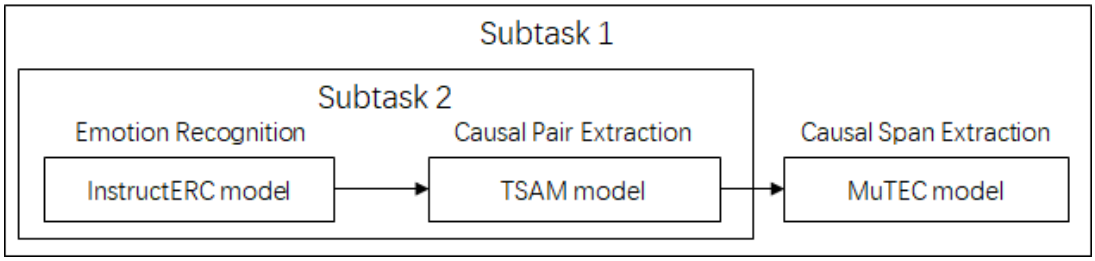


Figure 1: The overview of proposed model framework.

3.2 Emotion Recognition in Conversations

3.2.1 InstructERC for Emotion Recognition

InstructERC (Lei et al., 2023b) reformulate the ERC task from a discriminative framework to a generative framework and design a prompt template which comprises job description, historical utterance window, label set and emotional domain retrieval module. Besides emotion recognition task, InstructERC also utilizes speaker identification and emotion prediction tasks for ERC task. The performance of emotional domain retrieval module, which is based on Sentence BERT (Reimers and Gurevych, 2019), rely on the abundance of corpus. Taking into account that no additional data can be used, we only retain job description, historical utterance window and label statement in the instruct template.

3.2.2 Hierarchical Emotion Label

The hierarchical classification structure is shown in Figure 2. The emotion labels in dataset can be split into three categories: neutral, positive and negative, which positive set consists of surprise and joy while negative set includes fear, sadness, disgust and anger.

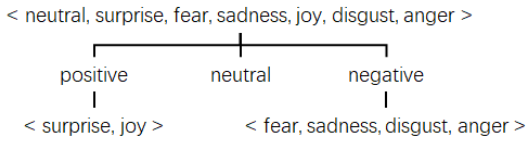


Figure 2: The Hierarchical Structure of Emotion labels.

3.2.3 Auxiliary Tasks and Instruct Design

Auxiliary tasks are proven as one of the efficient data augment methods (Lei et al., 2023b). Besides emotion recognition and speaker identification tasks, we add three auxiliary tasks in training data: sub-label recognition, positive recognition, and negative recognition tasks. The instruct template is depicted in Figure 3.

For emotion recognition and speaker identification task, we follow the format of instruct template in InstructERC, which consists of job description, historical content and label statement. For sub-label recognition (SR), positive recognition (PR) and negative recognition (NR) tasks, we utilize the corresponding label set which is mentioned in Section 3.2.2 to replace the label statement separately. The number of Speakers in the dataset is 304. The number of utterances from other speakers except the protagonist is far lower than the number of protagonists. Therefore, we unified all speakers other than the protagonist into 'Others'.

Visual data also plays an essential role in ERC. For video clips, we utilize LLaVA to generate descriptions of background, speaker movement and personal state. Therefore, we add background description, movement description and personal state description in instruct template. The background exhibits the information of scene in the conversation. The movement description depicts the action of speakers during corresponding utterances. The personal state description provides the observation of speakers' facial expressions. Considering the influence of the context, we have generated two sets of descriptions. The input of the first group only includes the clips corresponding to the utterances, while the second group adds the clips sequence corresponding to the historical utterances to the input of second group.

3.3 Emotion Cause Span Extraction

Emotion cause span extraction aims to extract the start position and end position of the causal utterance in a conversation. Typically, we can utilize a pipeline framework which firstly predicts the emotion and then predicts the cause span. For the cause span predictor, we can use SpanBERT (Joshi et al., 2020), RoBERTa (Liu et al., 2019) as the feature extractor and employ two heads on the top of them to extract the start and end positions given the causal

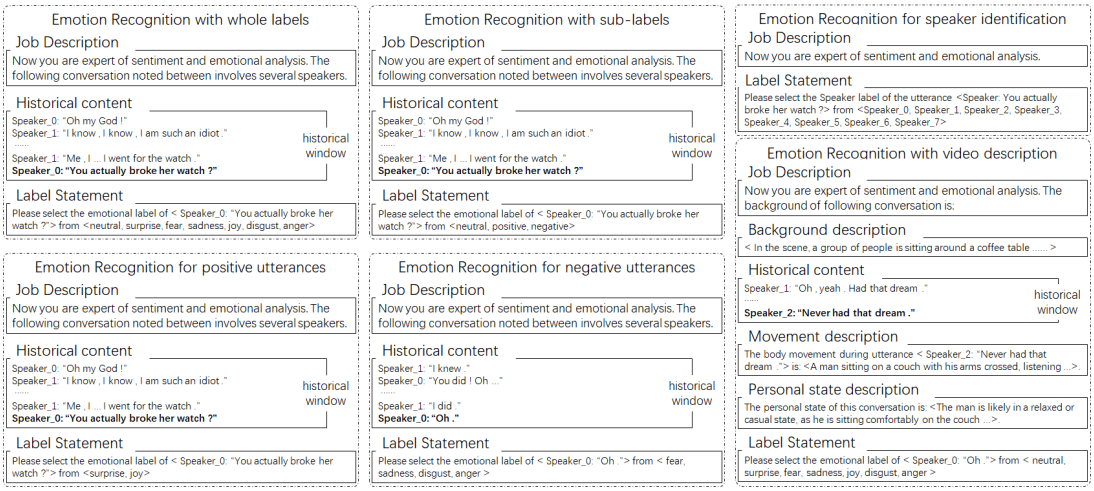


Figure 3: The Schematic of Instruct Template for ERC.

utterance. The two-step model offers an advantage in its modularity, allowing the application of distinct architectures for the emotion predictor and cause span predictor. However, it comes with two drawbacks: 1) Errors in the first step can propagate to the next, and 2) This approach assumes that emotion prediction and cause-span prediction are mutually exclusive tasks. In our system, we follow MuTEC Bhat and Modi (2023) and use an end-to-end framework in a joint multi-task learning manner to extract the causal span in a conversation.

During the training period, the input comprises the target utterance U_t, the candidate causes utterance U_i, and the historical context. MuTEC employs a pre-trained model (PLM) to extract the context representations. For emotion recognition, which is an auxiliary task, it employs a classification head on the top of the PLM. The end position is predicted by the prediction head of the concatenated representations of the given start index and the sequence output from the PLM. In this stage, the golden start index is used as the start index. The training loss is a linear combination of the loss for cause-span prediction and emotion prediction: $\mathcal{L}_{Loss} = \mathcal{L}_{CSE} + \beta\mathcal{L}_{Emotion}$.

During the inference period, as the start index is unknown, it uses top k start indices as the candidate start indices and gets k candidate end indices. Finally, it gets the final start-end indices by argmaxing the $k \times k$ start-end pairs.

3.4 Emotion-Cause Pair Extraction

3.4.1 TSAM Model

In our pipeline framework, for Subtask2, we first extract the emotion of the utterance and then extract the causal pairs given the emotional utterance in a conversation. The causal pairs extraction is typically modelled as the causal emotion entailment (CEE) task. In our system, we employ TSAM model from Zhang et al. (2022) as the causal pair extractor. TSAM mainly comprises three modules: Speaker Attention Network (SAN), Emotion Attention Network (EAN), and Interaction Network (IN). The EAN and SAN integrate emotion and speaker information simultaneously, and the subsequent interaction module efficiently exchanges pertinent information between the EAN and SAN through a mutual BiAffine transformation (Dozat and Manning, 2016).

Contextual Utterance Representation The pre-trained RoBERTa is employed as the utterance encoder, and we obtain contextual utterance representations by inputting the entire conversational history U_t, into the RoBERTa (Liu et al., 2019), separated by a special token [CLS], where $i = 0, 1, 2, ..., t$. We use the representation of [CLS] as the contextual representation of the utterance, which can be denoted as $h_u^i \in H_u$.

Emotion Attention Network To represent emotions, the EAN utilizes an emotion embedding network as the extractor of emotion representations, $X_e^k = Embedding(e_k)$, where e_k represents k-th emotion label. The embedding network can be considered as the lookup-table operation. The emotion embedding matrix is initialized using a random initializer and is fine-tuned throughout the training process. Employing a multi-head attention mechanism (Devlin et al., 2018), the EAN treats utterance representations as query vectors and emotion

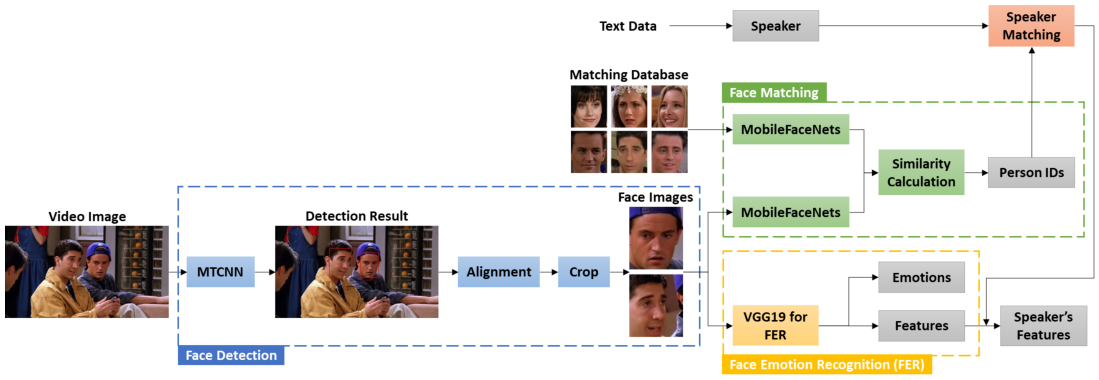


Figure 4: The framework of the face module.

representations as key and value vectors. The calculation process of the EAN mirrors that of a typical multi-head self-attention module (MHSA).

$$H_e = MHSA(Q, K, V) \tag{1}$$

where $Q = H_u, K = V = H_e$.

Speaker Attention Network The SAN facilitates interactions between utterances to incorporate speaker information by applying attention over the speaker relation graph. There are two types of relation edges: (1) Intra-relation type, which signifies how the utterance influences other utterances, including itself, expressed by the same speaker; (2) Inter-relation type, indicating how the utterance influences those expressed by other speakers. The speaker representation given a relationship can be formulated by the graphical attention mechanism (Zhang et al., 2022).

$$\begin{aligned} h_s^i &= \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \alpha_{ijr} W_r h_j^u \\ \alpha_{ijr} &= softmax(ReLu(\alpha_r^T W_r [h_i^u || h_j^u])) \end{aligned} \tag{2}$$

Interaction Network To efficiently exchange pertinent information between the EAN and SAN, a mutual Bi-Affine transformation is applied as a bridge (Dozat and Manning, 2016). In our Interaction Network, we integrate a masking mechanism to accommodate the existence of empty utterance speakers in some instances, which differs from the original approach. We denote this approach as the Masking Interaction Network (MIN).

$$\begin{aligned} \dot{H}_e &= softmax(Mask(H_e W_1 H_s^T)) H_s \\ \dot{H}_s &= softmax(Mask(H_s W_2 H_e^T)) H_e \end{aligned} \tag{3}$$

Cause Predictor The ultimate utterance representation for U_i is acquired by concatenating the output $\dot{H}_e$ and $\dot{H}_s$ from the L-layer TSAM. Subsequently, the concatenated vector undergoes classification using a fully-connected network. Given the target utterance U_i, the causal probability of the U_j can be formulated as follows:

$$p_{i,j} = sigmoid(fc(H_s^j || H_e^j)) \tag{4}$$

Multi-task Learning Auxiliary Task (MTLA) One drawback of the pipeline framework is that the extraction of utterance emotion and causal information are treated as separate tasks, potentially limiting the exploration of implicit relationships between them. Therefore, we incorporate emotion prediction as an auxiliary task within a multi-task learning framework. For emotion prediction, we utilize a classification head atop the Transformer-based model and apply the Dice loss (Li et al., 2019) as the multi-category classification loss.

3.5 Infusion of Video and Audio Information

The video data potentially carries rich knowledge for emotion analysis and existing research (Caridakis et al., 2007) has underscored the significance of multi-modal information in augmenting the semantic prediction capabilities of models. Our study leverages the visual and auditory cues present in conversational contexts with the aim of bolstering the efficacy of our language models in emotion analysis tasks.

3.5.1 Embedding and Concating Strategy

We set up specific embedding and fusion strategies for different language models. For BERT, we use the concatenation of textual and multi-modal features in the hidden layer. For Large Language Models (LLMs), our approach is characterized by the utilization of visual captions as supportive prompts, thereby furnishing the LLMs with an enriched informational context.

Models	LLM	w-avg F1	Accuracy
Origin InstructERC	Llama-2-7B-chat	53.83	50.87
Origin InstructERC	Llama-2-13B-chat	55.50	48.93
Ours-ERC-7B	Llama-2-7B-chat		
+ 3 auxiliary tasks		56.88	61.38
+ 3 auxiliary tasks & historical clips desc		57.74	57.02
+ 3 auxiliary tasks & utterance clips desc		58.42	57.92
Ours-ERC-13B	Llama-2-13B-chat		
+ 3 auxiliary tasks		57.85	**61.45**
+ 3 auxiliary tasks & historical clips desc		**58.64**	60.83
+ 3 auxiliary tasks & utterance clips desc		58.50	61.04

Table 1: Results of ERC task on test set without neutral utterances.

3.5.2 Extract Audio Feature Set

Audio data contains valuable information for emotion analysis, including tone, pitch, speed, and intensity of speech, as well as non-linguistic sounds and pauses, which together convey rich emotional cues. We use openSMILE (Eyben et al., 2010) to extract two comprehensive feature sets: GeMAPS (Eyben et al., 2016) and ComParE (Schuller and Batliner, 2013). GeMAPS is proposed for its effectiveness in capturing emotion-relevant vocal characteristics and ComParE encompasses a wide range of descriptors.

3.5.3 Video Image to Text

Integrating multi-modal features directly into the hidden layers of Large Language Models (LLMs) presents a significant challenge, primarily due to the prohibitive requirements for data and computational resources, such as GPUs. Although some finetuning strategies like prompt tuning could achieve it by addiing features to the input layer, we convert video to text with captioning where we can leverage our well-trained ERC model.

The performance of image captioning has been further enhanced with the outstanding NLU ability of LLMs. Large VLMs like LLaVA (Liu et al., 2023a) provide GPT-4 level multi-modal capability by visual instruction tuning. Furthermore, the Audio-Visual Language Model, Video-Llama (Zhang et al., 2023a), integrates both visual and audio encoders, enabling the comprehensive fusion of entire video content into LLMs. Without further training the VLMs as lack data, a well-designed prompt instructs the model to generate an emotion-related description. Our prompt asks the model to generate information from the front-ground event and place to character movements, the main character, facial expression, and finally emotion. The use of Chain-of-Thought (Wei et al., 2022) prompting further guides the model through a step-by-step process to derive the final emotion label. The output generated at each step is then incorporated into the ERC model, enriching it with a more detailed informational context.

3.5.4 Video image to Face Embedding

The faces in the video images contain rich emotion-related information, so pre-trained models are used to extract the face embeddings and correspond the identity of the face to the speaker in the text. The framework of the face module is shown in Figure 4.

Firstly, the Multi-Task Convolutional Neural Network (MTCNN) (Zhang et al., 2016) is used to detect the bounding boxes and key points of the faces. Next, the face images are affine transformed to a forward and intermediate state, and the faces are cropped and resized. The cropped images are then used for two subtasks: face matching and Face Emotion Recognition (FER). During face matching, two images of each protagonist are selected to build a matching database. With the help of MobileFaceNets (Chen et al., 2018), the embeddings of the face images are extracted, and the identity of each face image is obtained by calculating its similarity with the embeddings of faces in the matching database. During FER, the emotion-related embedding of the face image corresponding to the speaker is extracted by VGG19 (Simonyan and Zisserman, 2015) for subsequent multimodal analysis. When the speaker is a supporting character that is not included in the matching database, the features of the face image with the largest area are selected. When no face is detected or the speaker cannot be matched, the output features are filled with 0.

3.6 Model Ensemble

Ensembling models has been proven to be effective in boosting system performance across various tasks (Zhang et al., 2023b). For the extraction of

Model	Pre-trained Model	Test Pos.F1*	Eval Pos.F1**
Origin TSAM	RoBERTa-base	74.3	-
Ours-CEE	base		
+MIN	RoBERTa-base	75.5	-
+MIN & MTLA	RoBERTa-base	75.9	-
+MIN & MTLA	RoBERTa-large	76.9	-
+MIN & MTLA & Ensemble	RoBERTa-large	**78.0**	**38.7**
Ours-CSE	BERT-base	-	31.62 (w-avg.)
Ours-CSE	RoBERTa-large	-	**32.23 (w-avg.)**

* The results are based on ground truth emotion labels.
** The results are based on emotion labels given by ERC.

Table 2: Results of our models for the causal emotion entailment subtask.

causal pairs, we utilize various models for ensemble learning. We utilize a majority voting mechanism to determine the final prediction, aiming for optimal performance on the test dataset.

4 Experimental Setup

4.1 Training Data

The split of dataset is same as SHARK (Wang et al., 2023b). The ECF dataset is divided into training, validation and test sets, which incclude 9966, 1087, 2566 utterances.

4.2 Training Details

For ERC task, we use InstructERC with Llama-2-7B-chat and LLamMA2-13B-chat, which retain default parameters. We finetune ERC model by peft on single A100 with batch size 8. The length of historical window is 12.

For both the causal emotion entailment subtask and the causal span extraction subtask, we adopt the default hyperparameter settings from the respective original papers. We found that conducting a hyperparameter grid search did not yield any additional performance improvements.

5 Results and Discuss

5.1 Emotion Recognition

We use weight average F1 score and accuracy to evaluate the performance of the model. It should be noted that according to the rules of the competition, we removed the neutral utterances when computing F1 score and accuracy. The result of ERC on test set is shown in Table 1. All models is trained on four auxiliary tasks mentioned by in Section 3.2.3. The best weight average F1 score is 58.64, which is achieved by Llama-2-13B with historical clips descriptions. The descriptions which contains information with the emotions of speakers improve 0.79 (from 57.85 to 58.64). As for accuracy, the Llama-2-13B without video clips descriptions achieves the highest score of 61.45. Compared with InstructERC's training data strategy, we have added additional auxiliary tasks and improve 12.52 on accuracy.

5.2 Emotion Cause Span Extraction

We utilize an end-to-end framework for cause span extraction and achieve a final performance of 32.23 in weighted average proportional F1 score on the official evaluation dataset as is shown in the Table 2. Our result significantly surpasses the result of 26.40 above $\sim +6.0$ achieved by the second-place participant. Furthermore, our results achieved the highest scores across all other official evaluation metrics, validating the effectiveness of our approach for subtask 1.

5.3 Causal Emotion Entailment

In our initial experiments focusing solely on text modality, we utilize the TSAM model as our baseline for the causal pair extraction subtask. As is shown in Table 2, After incorporating the MIN, our positive F1 score improves by $+1.2$. Furthermore, with the introduction of emotional multi-task learning as an auxiliary task, our result sees an additional improvement of $+0.4$. Furthermore, we achieve an additional improvement of approximately $\sim +1.1$ in the official final evaluation dataset through model ensembling.

We also conduct experiments involving other modalities, including audio and vision, as is show in Table 3. For both audio and vision features, we concatenate them with the pure textual features. Regarding audio, we experiment with two public feature sets: GeMAPS and ComParE. The GeMAPS feature has a dimension of 62, while the ComParE

Modality	Feature Set	Feature Selection	Feature Dimension	Test Pos.F1
Audio	GeMAPS	×	62	39.0
	ComParE	×	top 1000	62.4
	ComParE	√	352	67.6
	ComParE	√	296	70.5
	ComParE	√	128	**73.9**
Vision	Max Img	×	128	70.7
	Speaker Img	×	128	74.3
	Emotional Speaker Img	×	512	**74.8**

Table 3: Results of multi-modality experiments for the causal emotion entailment subtask.

feature has a dimension of 6373. For the ComParE features, we employ an L1-based logistic regression classifier for feature selection, and we find that the best performance is achieved with a feature selection dimension of 128, resulting in a performance of 73.9. For the vision modality, we achieve a performance of 74.8, which is comparable to the result of the audio modality. However, upon introducing either audio or visual modalities, we observe a decreasing trend compared to the pure textual modality. This observation inspires us to develop a more reasonable approach to incorporate multi-modality in conversation analysis.

6 Conclusion

In this paper, we propose a joint pipeline framework for Subtask1 and Subtask2. Firstly, we utilize the Llama-2-based Instruct ERC model to extract the emotional content of utterances in a conversation. Next, we employ a two-stream attention model to identify causal pairs based on the predicted emotional states of the utterances. Lastly, we adopt an end-to-end framework using a multi-task learning approach to extract causal spans within a conversation. Our approach achieved first place in the competition, and the effectiveness of our approach is further confirmed by the ablation study. In future work, we plan to explore the integration of audio and visual modalities to enhance the performance of the task.

References

Ashwani Bhat and Ashutosh Modi. 2023. Multi-task learning framework for extracting emotion cause span and entailment in conversations. In *Transfer Learning for Natural Language Processing Workshop*, pages 33–51.

Busso C., Bulut M., and Lee et al. CC. 2008. IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

George Caridakis, Ginevra Castellano, Loic Kessous, Amaryllis Raouzaiou, Lori Malatesta, Stelios Asteriadis, and Kostas Karpouzis. 2007. Multimodal emotion recognition from expressive faces, body gestures and speech. In *Artificial Intelligence and Innovations 2007: from Theory to Applications*, pages 375–388, Boston, MA. Springer US.

Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. 2018. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *Biometric Recognition: 13th Chinese Conference, CCBR 2018, Urumqi, China, August 11-12, 2018, Proceedings 13*, pages 428–438. Springer.

Vishal Chudasama, Purbayan Kar, Ashish Gudmalwar, Nirmesh Shah, Pankaj Wasnik, and Naoyuki Onoe. 2022. M2fnet: Multi-modal fusion network for emotion recognition in conversation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4652–4661.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *In Association for Computational Linguistics (ACL)*, page 3161–3170.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Florian Eyben, Klaus R. Scherer, Björn W. Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong. 2016. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2):190–202.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1459–1462, New York, NY, USA. Association for Computing Machinery.

Chuang Fan, Chaofa Yuan, Jiachen Du, Lin Gui, Min Yang, and Ruifeng Xu. 2020. Transition-based directed graph construction for emotion-cause pair extraction. In *In Association for Computational Linguistics (ACL)*, page 3707–3717.

Bernhard Freudenthaler, Jorge Martinez-Gil, Anna Fensel, Kai Höfig, Stefan Huber, and Dirk Jacob. 2022. KI-Net: Ai-based optimization in industrial manufacturing—a project overview. In *International Conference on Computer Aided Systems Theory*, pages 554–561. Springer.

Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. Cosmic: Commonsense knowledge for emotion identification in conversations. *arXiv preprint arXiv:2010.02795*.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. *arXiv preprint arXiv:1908.11540*.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018. ICON: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604.

Dou Hu, Yinan Bao, Lingwei Wei, Wei Zhou, and Songlin Hu. 2023. Supervised adversarial contrastive learning for emotion recognition in conversations. *arXiv preprint arXiv:2306.01505*.

Guimin Hu, Ting-En Lin, Yi Zhao, Guangming Lu, Yuchuan Wu, and Yongbin Li. 2022. UniMSE: Towards unified multimodal sentiment analysis and emotion recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7837–7851.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Taewoon Kim and Piek Vossen. 2021. EmoBERTa: Speaker-aware emotion recognition in conversation with roberta.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023a. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework. *arXiv preprint arXiv:2309.11911*.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023b. InstructERC: Reforming emotion recognition in conversation with a retrieval multi-task llms framework. *arXiv preprint*, arXiv:2309.11911.

Shanglin Lei, Xiaoping Wang, Guanting Dong, Jiang Li, and Yingjian Liu. 2023c. Watch the speakers: A hybrid continuous attribution network for emotion recognition in conversation with emotion disentanglement. In *2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 881–888.

Jiang Li, Xiaoping Wang, Guoqing Lv, and Zhigang Zeng. 2024. GraphCFC: A directed graph based cross-modal feature complementation approach for multimodal conversational emotion recognition. *IEEE Transactions on Multimedia*, 26:77–89.

Jiangnan Li, Fandong Meng, Zheng Lin, Rui Liu, Peng Fu, Yanan Cao, Weiping Wang, and Jie Zhou. 2022. Neutral utterances are also causes: Enhancing conversational causal emotion entailment with social commonsense knowledge. *arXiv preprint arXiv:2205.00759*.

Jingye Li, Meishan Zhang, Donghong Ji, and Yijiang Liu. 2020. Multi-task learning with auxiliary speaker identification for conversational emotion recognition. *arXiv preprint arXiv:2003.01478*.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog a manually labelled multi-turn dialogue dataset. *arXiv preprint*, arXiv:1710.03957.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

Xiao Liu, Jian Zhang, Heng Zhang, Fuzhao Xue, and Yang You. 2023b. Hierarchical dialogue understanding with special tokens and turn-level attention. *arXiv preprint arXiv:2305.00262*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, and E. Cambria. 2019. DialogueRNN: An attentive rnn for emotion detection in conversations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6818–6825.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the*

55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 873–883.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint*, arXiv:1810.02508.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, et al. 2021. Recognizing emotion cause in conversations. *Cognitive Computation*, 13:1317–1332.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Bjorn Schuller and Anton Batliner. 2013. *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*, 1st edition. Wiley Publishing.

Weizhou Shen, Junqing Chen, Xiaojun Quan, and Zhixian Xie. 2021a. DialogXL: All-in-one xlnet for multi-party conversation emotion recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13789–13797.

Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 2021b. Directed acyclic graph network for conversational emotion recognition. *arXiv preprint arXiv:2105.12907*.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Xiaohui Song, Longtao Huang, Hui Xue, and Songlin Hu. 2022. Supervised prototypical contrastive learning for emotion recognition in conversation. *arXiv preprint arXiv:2210.08713*.

Ishiwatari Taichi, Yasuda Yuki, Miyazaki Taro, and Goto Jun. 2020. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7360–7370. Association for Computational Linguistics.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023a. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Fanfan Wang, Jianfei Yu, and Rui Xia. 2023b. Generative emotion cause triplet extraction in conversations with commonsense knowledge. In *In Findings of the Association for Computational Linguistics: EMNLP 2023*, page 3952–3963.

Yan Wang, Jiayu Zhang, Jun Ma, Shaojun Wang, and Jing Xiao. 2020. Contextualized emotion recognition in conversation as sequence tagging. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 186–195.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Penghui Wei, Jiahao Zhao, and Wenji Mao. 2020. Effective inter-clause modeling for end-to-end emotion-cause pair extraction. In *In Association for Computational Linguistics (ACL)*, page 3171–3181.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Sayyed M. Zahiri and Jinho D. Choi. 2017. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. *arXiv preprint*, arXiv:1708.04299.

Dong Zhang, Liangqing Wu, Changlong Sun, and et.al. 2019. Modeling both context-and speaker-sensitive dependence for emotion detection in multi-speaker conversations. In *IJCAI*, pages 5415–5421.

Duzhen Zhang, Zhen Yang, Fandong Meng, Xiuyi Chen, and Jie Zhou. 2022. Tsam: A two-stream attention model for causal emotion entailment. *arXiv preprint arXiv:2203.00819*.

Hang Zhang, Xin Li, and Lidong Bing. 2023a. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

Haojie Zhang, Xiao Li, Renhua Gu, Xiaoyan Qu, Xiangfeng Meng, Shuo Hu, and Song Liu. 2023b. Samsung research china-beijing at semeval-2023 task 2: An al-r model for multilingual complex named entity recognition. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 114–120.

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503.

Weixiang Zhao, Yanyan Zhao, Zhuojun Li, and Bing Qin. 2023. Knowledge-bridged causal interaction network for causal emotion entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):14020–14028.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 165–176.

Lixing Zhu, Gabriele Pergola, Lin Gui, Deyu Zhou, and Yulan He. 2021. Topic-driven and knowledge-aware transformer for dialogue emotion detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1571–1582.

Werkzeug at SemEval-2024 Task 8: LLM-Generated Text Detection via Gated Mixture-of-Experts Fine-Tuning

Youlin Wu*, **Kaichun Wang***, **Kai Ma, Liang Yang, Hongfei Lin**†
School of Computer Science and Technology, Dalian University of Technology, China
{wuyoulin, wangkc, ostmbh}@mail.dlut.edu.cn
{liang, hflin}@dlut.edu.cn

Abstract

Recent advancements in Large Language Models (LLMs) have propelled text generation to unprecedented heights, approaching human-level quality. However, it poses a new challenge to distinguish LLM-generated text from human-written text. Presently, most methods address this issue through classification, achieved by fine-tuning on small language models. Unfortunately, small language models suffer from anisotropy issue, where encoded text embeddings become difficult to differentiate in the latent space. Moreover, LLMs possess the ability to alter language styles with versatility, further complicating the classification task. To tackle these challenges, we propose **G**ated **M**ixture-**o**f-**E**xperts **F**ine-tuning (GMoEF) to detect LLM-generated text. GMoEF leverages parametric whitening to normalize text embeddings, thereby mitigating the anisotropy problem. Additionally, GMoEF employs the mixture-of-experts framework equipped with gating router to capture features of LLM-generated text from multiple perspectives. Our GMoEF achieved an impressive ranking of #8 out of 70 teams. The source code is available on https://gitlab.com/werkzeug1/gmoef.

1 Introduction

The advancements in Large Language Models (LLMs) have made generating human-level text more accessible and cost-effective than ever before. These advancements, coupled with techniques such as chain-of-thought (Wei et al., 2022) and instruction tuning (Zhang et al., 2023), have enabled LLMs in producing high-quality text on various topics. However, in the real world, using LLM-generate text is not always acceptable. Thus, there is an urgent need for an easy yet reliable way to detect LLM-generated text.

The SemEval-2024 task 8 (Wang et al., 2024) aims to find methods that can detect machine-generated text. In this work, we followed the most common black-box detection paradigm, which regards such problem as a classification task. We argue that current methods all suffer from the following issues: (1) Anisotropy of the text embeddings (Li et al., 2020; Jiang et al., 2022; Gao et al., 2021). Using small pretrained language models (PLMs) to encode text is the very first step for all classification models, however, PLM may suffer from anisotropy issue, which makes text embeddings clustering in a small cone in the latent space, and compromise the classification performance. (2) Language style of LLM-generated text is dynamic. As aforementioned, LLM can generate text that accommodates various topics and contexts; different LLM may have different optimization targets during pre-training *w.r.t.* text generation. In other words, finding a regular pattern for LLM-generated text is difficult.

To this end, we propose **G**ated **M**ixture-**o**f-**E**xperts **F**ine-tuning (GMoEF) to tackle these problems. GMoEF first uses the PLM to encode the text, then employs parametric whitening transformation to normalize the embedding distribution, in order to mitigate the anisotropy issue; furthermore GMoEF adopts Mixture-of-Experts equipped with gating router to capture features of LLM-generated text from multiple perspectives. Our GMoEF achieved an impressive ranking of #8 out of 70 participating teams on subtask B.

2 Related Work

Typically, LLM-generated text detection is regarded as a classification task aimed at distinguishing between LLM-generated text and human-written text (Jawahar et al., 2020). With the advancement of LLMs, their text generation capabilities have reached a level comparable to hu-

*Equal contribution.
†Corresponding author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 547–552

man writing (Achiam et al., 2023), making it even challenging for humans to differentiate between LLM-generated text and human-written text. Consequently, there is a need to develop effective detectors to mitigate the potential misuse of LLM (Wu et al., 2023). Recently, owing to the construction of numerous high-quality benchmarks and innovations in detection methods, significant progress has been made in LLM-generated text detection technology.

High-quality datasets play a crucial role in advancing research on detecting LLM-generated text. HC3 dataset (Guo et al., 2023), represents one of the pioneering open-source efforts aimed at comparing ChatGPT-generated text with human-written text.The CHEAT dataset (Yu et al., 2023) comprises academic abstracts written by humans sourced from IEEE Xplore, and is committed to detecting artificially generated deceptive academic content from ChatGPT. Additionally, there are numerous datasets containing text generated by various LLMs, such as monolingual datasets DeepfakeText-Detect-Dataset (Li et al., 2023), GPT-written dataset (Liu et al., 2023b), and M4 (Wang et al., 2023), used in this competition.

Focusing on recently proposed detection methods, these primarily encompass zero-shot (Corston-Oliver et al., 2001), fine-tuning LMs (Qiu et al., 2020), adversarial learning (Hu et al., 2023), and LLMs as detectors (Koike et al., 2023). DetectGPT (Mitchell et al., 2023) is dedicated to the detection of LLM-generated text by analyzing the structural attributes inherent in the probability functions of LLMs. Fagni et al. (2021) noted that fine-tuning RoBERTa (Liu et al., 2019a) resulted in optimal classification outcomes across diverse encoding configurations. Recent studies (Liu et al., 2023a; Chen et al.) have additionally supported the outstanding performance of fine-tuned variants within the BERT family, such as RoBERTa, in discerning LLM-generated text. Yang et al. (2023) conducted an adversarial data augmentation process on LLM-generated text, and the results showed that models trained with augmented data exhibited enhanced robustness.

3 Methodology

In this section, we present our GMoEF in details. We first introduce the overall architecture of the proposed GMoEF, then give a comprehensive insight of the adopted parametric whitening and gated mixture-of-experts

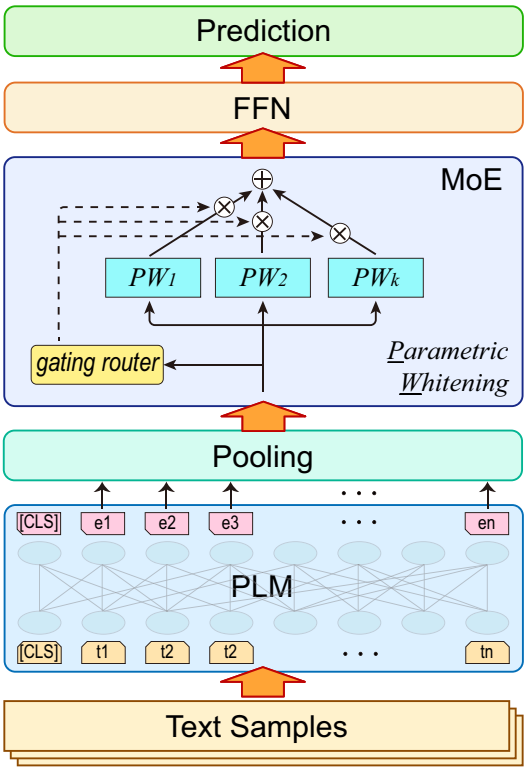


Figure 1: The main architecture of GMoEF.

3.1 System Architecture

The overall architecture is shown in Figure 1. Basically, our GMoEF follows the fine-tuning PLM as the classifier paradigm. We first employ a PLM as the text encoder. For text sample s_i, we take the last layer output at each token position through a mean pooling layer to obtain the text embedding x_i. Notably, we do not take the commonly adopted `[CLS]` position output as the text embedding. Further discussion can be found in section 4.3. On acquiring the text embedding, we put it through a gated mixture-of-expert layer, in which we adopt parametric whitening module as the expert, to learn the language feature of LLM-generated text. Finally, we employ a feed-forward network to give the final probability score $\widehat{y}_i$. We then use the cross-entropy loss as the optimization target.

$$\mathcal{L} = -\sum_{i=1}^{k} y_i \log(\widehat{y}_i) + (1 - y_i) \log(1 - \widehat{y}_i). \quad (1)$$

3.2 Parametric Whitening

While we can utilize a PLM to encode texts into embeddings, current studies have revealed that PLMs induce a non-smooth, anisotropic semantic space

for general texts(Li et al., 2020). Anisotropy issue makes the embeddings occupy a narrow cone, resulting in a high similarity between any embedding pairs. Consequently, this situation can have a negative impact on downstream classification tasks (Jiang et al., 2022). The problem is further exacerbated when mixing texts generated by multiple LLMs and written by humans. Drawing inspiration from recent studies that aim to improve PLM-generated text embeddings through whitening-based methods (Su et al., 2021; Huang et al., 2021), we incorporate a simple linear transformation to transform the original PLM generated embeddings for deriving isotropic representations. Unlike previous whitening-based methods, we make mean and variance as two learnable parameters, for better generalizability. We define the whitening transformation as:

$$\widetilde{x_i} = (x_i - \mathbf{b}) \cdot \mathbf{W}_1, \tag{2}$$

where $x_i \in \mathbb{R}^d$ is the original text embedding, while $\mathbf{b} \in \mathbb{R}^d$ and $\mathbf{W}_1 \in \mathbb{R}^{d \times d'}$ are all parameters to learn. $\widetilde{x_i}$ is the transformed text embedding.

3.3 Gated Mixture-of-Experts

As mentioned earlier, the dynamic language style of LLM-generated text poses a significant challenge for all detecting methods. We contend that conventional methods are only capable of capturing limited or partial aspects of the pattern. To this end, we employ multiple parametric whitening layers to learn a series of whitening embeddings. Each embedding will focus on a certain aspect of the language style, and we make the final decision based on all these embeddings to draw a more robust conclusion.

To implement our idea, we employ the mixture-of-experts (MoE) architecture (Jacobs et al., 1991; Eigen et al., 2013). More sepcifically, we employ k parametric whitening layers as the *experts*, then employ a gating router (Shazeer et al., 2016; Hou et al., 2022) to aggregate them. For text sample s_i, the gated mixture-of-expert output v_i is defined as:

$$v_i = \sum_{j=1}^{k} g_j \widetilde{x_i}^{(j)}, \tag{3}$$

where $\widetilde{x_i}^{(j)}$ represents j-th whitening transformed embedding for text sample s_i. g_j is the weight derived from the gating router, which is defined as follows:

$$\boldsymbol{g} = \text{Softmax}(x_i \cdot \mathbf{W_2} + \boldsymbol{\delta}), \tag{4}$$

$$\boldsymbol{\delta} = \boldsymbol{\epsilon} \cdot \text{Softplus}(x_i \cdot \mathbf{W_3}). \tag{5}$$

where $\boldsymbol{g} \in \mathbb{R}^k$ is the routing vector. We employ two learnable parameters $\mathbf{W_2}$ and $\mathbf{W_3}$ to dynamically adjust the weight for each expert. Inspired by Inoue (2019), we incorporate a series of noises $\boldsymbol{\delta}$ in the gating router to balance these experts and avoid overfitting.

4 Experiment

4.1 Experimental setup

Dataset and Evaluation. A sampled version of M4 (Wang et al., 2023) dataset provided by the organizer was adopted. Comprehensive statistics regarding the dataset can be found in Table 1. Subtask A focuses on detecting single-model generated text while subtask B focuses on the multi-model generated text distinguish. However, subtask C has a very different optimization target comparing to subtask A and B, we opted not to conduct experiments on this particular subtask. As mentioned in the official task description, we employed Accuracy as the evaluation metric to assess the quality of the detection.

Subtask	#Train	#Dev	#Test
A (mono.)	119,757	5,000	34,272
A (multi.)	172,417	4,000	42,378
Subtask B	71,027	3,000	18,000

Table 1: Statistics on subtask A (monolingual & multilingual) and subtask B.

Implementation details. We implemented the GMoEF model based on RoBERTa[1] (Liu et al., 2019b) and XLM-R[2] (Conneau and Lample, 2019) for monolingual and multilingual scenarios respectively, with Pytorch (Paszke et al., 2019) and the Huggingface Transformers library (Wolf et al., 2020). To facilitate distributed training, we utilized the pytorch-lightning framework (Falcon and The PyTorch Lightning team, 2019).

For optimization, we used the AdamW optimizer with an initial learning rate of $2e^{-4}$ for the RoBERTa part and $2e^{-5}$ for the non-RoBERTa parts. The learning rate was linearly decayed with 10% warm-up steps. The hyperparameter settings

[1] https://huggingface.co/FacebookAI/roberta-large
[2] https://huggingface.co/FacebookAI/xlm-roberta-large

Hyperparameter	Symbol	Value
Maximum words (tokens)	-	512
# of experts	k	3
# of epochs	-	3
weight decay	-	$1e^{-2}$
seed	-	42
batch size	-	$32^{\dagger}$
hidden dim	d'	256
PLM embedding dim	d	1024

Table 2: The hyperparameters of the experiment. †: on a single GPU.

we employed are summarized in Table 2. All models are trained on two NVIDIA-SXM4-A100 GPUs.

4.2 Main Results

The main results on test set are shown in Table 3. Our GMoEF exhibits impressive results in both subtask A and subtask B. However, our original submissions (*orig. sub.*) for subtask A is not satisfying as expected. We attribute this discrepancy as two folds: 1) It is possible that we failed to identify the optimal checkpoint for generating predictions on the test set due to a substantial disparity between the number of training and evaluation samples. 2) We searched for the optimal number of experts (k) from 4 up to 10 during submission stage, however, the best result shows up at $k = 3$.

We further find out that our GMoEF shows more significant performance improvements on subtask A (multilingual) and subtask B over the baselines. However, interestingly, the GMoEF does not exhibit significant advantages in subtask A (monolingual). It may indicates that the GMoEF is better suited for complex scenarios, for instance, the texts are *multilingual* and may generated by *multiple models*.

4.3 Ablation Study

In order to validate the unique contribution of each module, we conduct experiments on the following variants of GMoEF:

- Without parametric whitening (*w/o* PW). In this variant, we substitute all parametric whitening layers into the linear layers.
- Using `[CLS]` position output as the text embedding (*alt.* PLM).

As shown in Table 3, all variants will lead to immediate performance drop on all subtasks, which further validates the necessity and effectiveness of all proposed model components. Through these results, we have several noteworthy observations: (1) The multilingual and multi-model cases exhibit more severe anisotropy issue. Removing the PW layer can lead to a substantial decline in performance. (2) The utilization of the `[CLS]` token for text encoding proves to be coarse-grained when it comes to capturing language styles or features in the LLM-generated text detection task. In this context, our token position pooling strategy emerges as a more suitable alternative.

Model	**A** (mono.)	**A** (multi.)	**B**
baseline	0.885	0.809	0.746
orig. sub.	0.806	0.768	0.822
GMoEF	**0.903**	**0.892***	**0.848***
improv.	2.03%	10.3%	13.7%
w/o PW	0.896	0.848	0.732
alt. PLM	0.845	0.808	0.711

Table 3: Experimental results on subtask A (monolingual & multilingual) and subtask B. The best results are marked in **boldface**. *w/o* stands for "without"; *alt.* stands for "alternative". "*" denotes that the improvements are significant at the level of 0.01 with paired *t*-test.

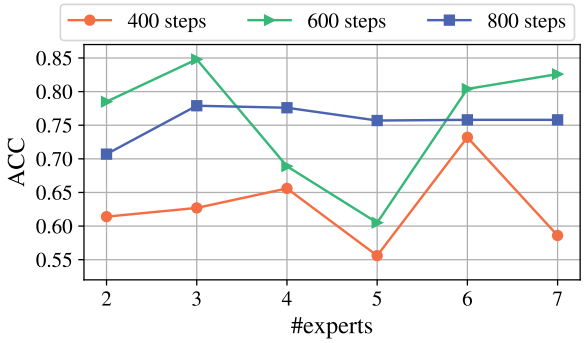


Figure 2: Experimental results on subtask B with different numbers of experts (k). Each line indicates start testing after training for certain steps. Notably, a whole training epoch takes ~1,110 steps under our setup.

4.4 Case Study on Number of Experts (k)

To reveal the effectiveness of our proposed gated mixture-of-experts fine-tuning, we further conduct experiments with different numbers of experts. Detailed results are shown in Figure 2, from these results, we have the following observations: (1) With the assistance of multiple experts, the model tends to converge much earlier, often requiring less

than a full epoch of training. The complete training process for subtask B takes about 1,110 steps. However, as shown in Figure 2, the optimal result is achieved at the 600th step with 3 experts. By the 800th step, the performance becomes expert-agnostic and suboptimal, indicating overfitting. (2) Our GMoEF achieves best performance with $k = 3$. With fewer experts, GMoEF can hardly capture the dynamic language features of LLM-generated text, and revert to conventional fine-tuning models. On the other hand, increasing the number of experts does not necessarily guarantee a better outcome. For instance, when $k = 5$, these experts may reach conflicting conclusions, leading to the worst result. While adding more experts may mitigate this phenomenon, it also introduces additional noise, ultimately resulting in suboptimal performance.

5 Conclusion and Future Work

In this work, we find out that current LLM-generated text detection methods may suffer from anisotropy issue, and they fail to capture the dynamic language features. To this end, we propose GMoEF, which incorporates parametric whitening to mitigate the anisotropy issue. GMoEF further adopts the Mixture-of-Experts equipped with gating router to model the pattern of LLM-generated text from multiple aspects. Our GMoEF exhibits an impressive #8 out of 70 participating teams on the multi-model generated text detection subtask. Extensive experiments show that our GMoEF is suitable for complicated scenarios where texts are multi-lingual and may generated by multiple possible LLMs.

In the future, we aim to extend our observations to other text classification tasks, and incorporate LLM itself to detect machine-generated text.

Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China (Grant No. 62376051, 62076046).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. Gpt-sentinel: Distinguishing human and chatgpt generated content.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Simon Corston-Oliver, Michael Gamon, and Chris Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *PLOS ONE*, page e0251415.

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. Whiteningbert: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244.

Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788*.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Ganesh Jawahar, Muhammad Abdul-Mageed, and LaksV.S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv: Computation and Language,arXiv: Computation and Language*.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. Promptbert: Improving bert sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023. Deepfake text detection in the wild. *arXiv preprint arXiv:2305.13242*.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023a. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *Cornell University - arXiv,Cornell University - arXiv*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023b. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, ChristopherD. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, page 1872–1897.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2016. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, et al. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F Wong, and Lidia S Chao. 2023. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724*.

Lingyi Yang, Feng Jiang, and Haizhou Li. 2023. Is chatgpt involved in texts? measure the polish ratio to detect chatgpt-generated text.

Peipeng Yu, Jiahan Chen, Xuan Feng, and Zhihua Xia. 2023. Cheat: A large-scale dataset for detecting chatgpt-written abstracts.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

SSN_Semeval10 at SemEval-2024 Task 10: Emotion Discovery and Reasoning its Flip in Conversations

Antony Rajesh A **Supriya Abirami A** **Chandrabose Aravindan** **Senthil Kumar B**

Department of Information Technology
Sri Sivasubramaniya Nadar College of Engineering
Chennai, Tamilnadu, INDIA
{*antony*2010532, *supriyaabirami*2010354, *AravindanC*, *Senthil*} @*ssn.edu.in*

Abstract

This paper presents a transformer-based classifier for recognizing emotions in Hindi-English code-mixed conversations, adhering to the SemEval task constraints. Leveraging BERT-based transformers, we fine-tune pre-trained models (mBERT and indicBERT) on the dataset, incorporating tokenization and attention mechanisms. Our approach achieved competitive performance (weighted F1-score of 0.4), showcasing the effectiveness of BERT in nuanced emotion analysis tasks within code-mixed conversational contexts. This F1-score was ranked 16^{th} among the 39 submissions.

1 Introduction

Recognition of emotions from conversation enables advancements in sentiment analysis, mental health monitoring, chatbot development and ultimately enhances user experiences and well-being. The EDiReF shared task (Task 10) at SemEval 2024(Kumar et al., 2024) comprises three subtasks: Emotion Recognition in Conversation (ERC)(Kumar et al., 2023) and Emotion Flip Reasoning (EFR)(Kumar et al., 2022) in both Hindi-English code-mixed conversations and English conversations. ERC involves assigning emotions to each utterance from a predefined set, while EFR aims to identify trigger utterances for emotion flips in multi-party conversations. This task is vital for understanding emotional dynamics in conversational contexts, particularly in multilingual settings like Hindi-English code-mixed conversations.

This paper proposes a classifier for ERC that adopts a BERT-based transformer architecture (Lee, 2022) for emotion recognition task. By fine-tuning pre-trained BERT models, like mBERT (DevlinJ et al., 2018) and indicBERT (Kakwani et al., 2020), on the given dataset, we leverage transfer learning to understand and reason about emotions effectively in multilingual conversational contexts like Hindi-English code-mixed conversations.

We participated in sub-task 1 (ERC) of Task 10 (EDiReF) and competed with 38 other teams within the provided time frame. Our system achieved rank 16 for this sub-task with a range of weighted F1-scores between 0.3 and 0.4 using BERT-based models. While we successfully utilized BERT-based models for emotion recognition in Hindi-English code-mixed conversations, our system encountered challenges in accurately capturing emotional contexts, which affected our overall performance.

2 Background

Sub-task 1 challenges participants to provide emotions as output for particular utterances in conversations. Both training and validation datasets are provided, with both datasets in textual format. The training set includes 343 conversations with 8505 utterances, while the validation set contains 46 conversations with 1354 utterances. Each conversation in both datasets comprises episodes, speakers, utterances, and emotions. Utterances are in Hindi-English code-mixed (e.g., *"Namaste, how are you?"*). The emotion distribution and utterance length distribution for both datasets are summarized in Figure 1 and Figure 2, respectively. Notably, the emotion distribution in both datasets is prominently skewed towards 'neutral', as indicated by the larger area in the distribution. Upon analysis, the emotions involved in both datasets are identified as *['neutral', 'contempt', 'sadness', 'fear', 'joy', 'surprise', 'anger', 'disgust']*.

3 Related Work

In recent years, emotion recognition in conversational contexts has seen significant contributions. (Maheshwari and Varma, 2022) focused on emo-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 553–557

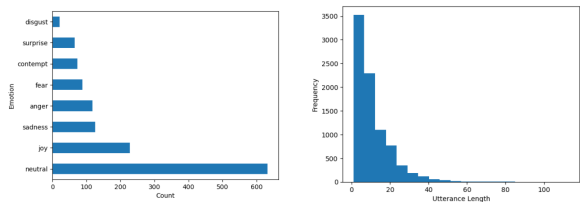


Figure 1: Emotion distribution and Utterance length distribution in training dataset

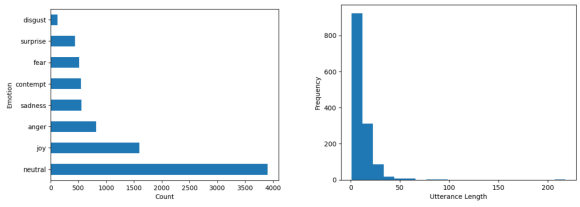


Figure 2: Emotion distribution and Utterance length distribution in validation dataset

tion recognition in tweets, emphasizing the importance of context. (Poria et al., 2019) survey offers a comprehensive overview of emotion recognition systems in dialogues, covering deep learning approaches and challenges. (Wang et al., 2023) study explores using transformers for emotion recognition in conversations, highlighting their effectiveness.

While deep learning has revolutionized the field, earlier works laid the foundation. (Thelwall et al., 2012) and (Pang and Lee, 2008) explored traditional approaches to Emotion Recognition (ER) using hand-crafted features and rule-based systems. (Mohammad and Turney, 2013) and (Tang et al., 2016) marked a shift towards deep learning for ER, focusing on Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for learning emotion representations from text data.

(Wadhawan and Aggarwal, 2021) introduces a new dataset for analyzing emotions in Hindi-English tweets and proposes a transformer-based approach using BERT to achieve state-of-the-art accuracy in emotion detection, outperforming other deep learning models like CNNs, LSTMs, Bi-LSTMs.

(D. et al., 2019) also contribute to the field by applying traditional and deep machine learning approaches to identify offensive language in social media, demonstrating the versatility of these techniques in analyzing online sentiment. This aligns with our work on emotion recognition in code-mixed social media data, as both studies explore methods for sentiment analysis in similar contexts.

And the recent case study, (Tatariya et al., 2024) mentions the challenges in code-mixed data for emotion classification. The study investigates the effectiveness of pre-trained language models in understanding sociolinguistic contexts. The findings underscore the importance of considering linguistic diversity and sociolinguistic factors in developing and interpreting emotion recognition models.

(Vijay et al., 2018) pioneered the work on emotion recognition in Hindi-English code-mixed social media text. Their work established a benchmark by creating a corpus of annotated data and proposing a classification system for emotion detection.

Building on Wadhawan and Aggarwal's success with BERT with the help of works done by (**?**) in SemEval 2021 and (Lee, 2022) in emotion recognition in conversations, our mBERT model aims to further improve emotion detection by addressing the cultural nuances and fine-tuning on a larger code-mixed hindi-english dataset while addressing the limitations highlighted by Tatariya et al.

4 System Overview

This section provides an overview of our BERT-based transformer system and justifies our selection of pre-trained BERT models. After data-preprocessing, our system takes conversations as input in form of sequence of tokens and produces emotion class as output for emotion classification. This processs is illustrated in Figure 3.

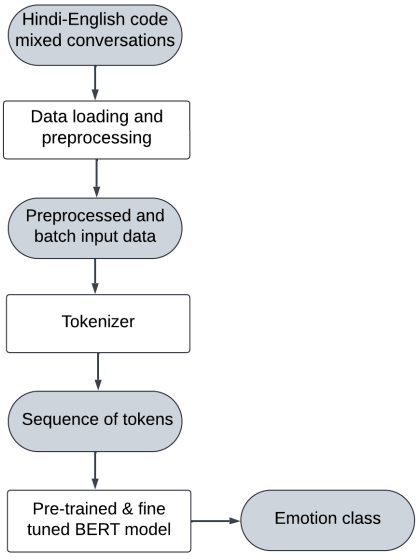


Figure 3: BERT-based Transformer System Overview

For Hindi-English Code-mixed language, we

used mBERT (DevlinJ et al., 2018) and indicBERT (Kakwani et al., 2020). Due to multilingual understanding of mBERT and IndicBERT which is designed for Indic languages, enabling them to process both Hindi and English as well as their mixtures effectively. Their cross-lingual transfer learning capabilities ensure robust performance with minimal fine-tuning, while their rich representations of language capture essential contextual information across language boundaries.

5 Experimental Setup

In this section, we present the implementation details of our system. During the training phase, we used the provided training dataset to fine-tune the BERT models and the validation set for evaluation. Later, both the training and the validation sets were used to fine-tune the BERT models which were submitted for testing using the test set.The stages involved in experiments are detailed below.

5.1 Data Pre-processing steps

5.1.1 JSON Parsing

To facilitate quick access to data samples, the given JSON dataset, containing information on episodes, speakers, utterances, and emotions, will be transformed into a text file with three columns: speakers, utterances, and emotions. A blank line in the text file will serve as the separator between different conversations.

5.1.2 Emotion loading for Specific Utterances and Retaining Previous Dialogue Context

Following JSON parsing, the data loader organizes input by loading the emotion associated with each dialogue alongside its utterances. Utterances undergo text cleaning, removing punctuations and stopwords in Hindi-English code-mixed languages. To grasp the current dialogue's emotion context, the loader loads the sequence of previous dialogues, including their utterances and speaker names. Speaker names are indexed starting from zero *(e.g., 0 for Ram, 1 for Divya)*, facilitating the mapping of utterances to their corresponding speakers. Additionally, the data loader maintains a set of emotions involved in the previous dialogue context.

5.2 Neural Architecture for emotion recognition in conversations

We downloaded the pre-trained mBERT[1] and indicBERT[2] models from the huggingface transformer library. We adopted the code of the transformer model for the Emotion Recognition Challenge (Lee, 2022) to implement our system. We processed the batch tokens through multiple layers of Transformer blocks, including self-attention mechanisms and feed-forward neural networks. For evaluating the performance metrics of every epoch while training, we used precision, recall, and weighted f1 score of the validation set. We used cross entropy loss for loss function and with the help of AdamW optimizer, we have updated the weights involved. The final layer of the BERT model determines the number of emotion classes using a data loader that tracks emotions in the input data, outputting the class label for classification. Class labels are then converted into emotions for analysis in the ERC task.

We conducted our experiments on Google Colab using the T4 GPU runtime mode. We trained the chosen BERT models with a batch size of 1 and a learning rate of 1e-6.

6 Results

We trained both the mBERT and indicBERT models according to the experimental setup. During training, we recorded and stored the weighted F1 scores of both the models on the validation set, which are detailed in Table 1. Notably, mBERT's performance improves until around 7-8 epochs, while indicBERT's score remains stable. Additionally, in Table 2, the final scores for precision, recall, and weighted F1 are detailed. Using the trained mBERT model, we achieved the 16th rank with a weighted F1 score of 0.4 in subtask 1 of task 10, whereas the top ranked system achieved a score of 0.78.

We examined the emotions predicted by two models, mBERT and indicBERT, and compared them to the actual test labels. We visualized the results using a confusion matrix of mBERT in Figure 4.

After normalizing the emotion distribution of the training dataset and mBERT correct predictions, we observed that their patterns appear similar in

[1]https://huggingface.co/google-bert/bert-base-multilingual-uncased

[2]https://huggingface.co/ai4bharat/indic-bert

Epochs	mBERT	indicBERT
1	29.44	28.34
2	35.35	29.1
3	35.49	29.1
4	38.68	29.1
5	40.72	29.1
6	41.4	29.12
7	41.56	29.18

Table 1: weighted f1 scores of mBERT and indicBERT for 7 epochs

Model	Precision (%)	Recall (%)	Weighted F1 Score (%)
M-BERT	41.46	47.56	42.47
IndicBERT	21.85	46.75	29.78

Table 2: mBERT and indicBERT - Final Performance metrics

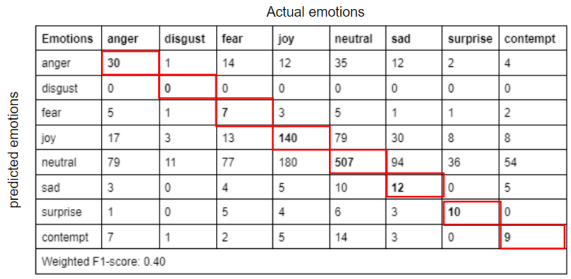


Figure 4: Confusion matrix with Highlighted correctly predicted emotions by mBERT

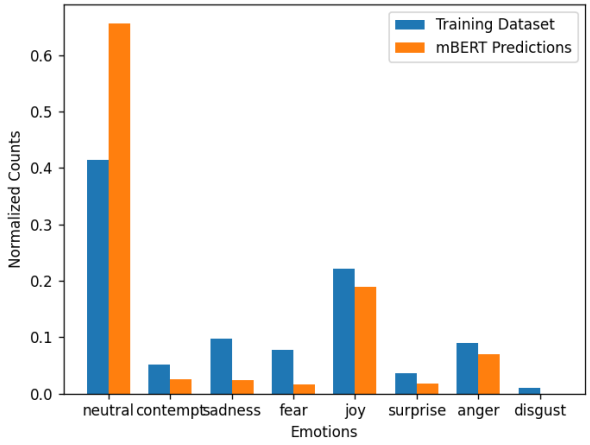


Figure 5: Normalized Emotion distribution of training dataset and mBERT correct predictions

Figure 5. However, *disgust*, *contempt*, *fear*, *sadness*, and *surprise* exhibit the lowest areas in the distribution, indicating that it is challenging for the model to identify utterances with these emotions. Therefore training datasets with a more balanced emotion distribution may possibly enhance the performance of mBERT.

The confusion matrix of indicBERT showed 0 for all the entries except for neutral where 656 test cases were predicted correctly. This clearly indicates that indicBERT has not learnt the contextual representations of utterances in the training dataset. This is primarily due to the fact that indicBERT was trained using Hindi unicode, whereas our dataset uses transliterated Hindi. We will try to resolve the issue in the future.

7 Conclusion

In our study on understanding emotions in Hindi-English conversations for SemEval 2024 Task 10, we used BERT-based models. Our system ranked 16th in subtask 1. However, accurately capturing nuanced emotions posed challenges, suggesting areas for improvement.

For future work, we plan to enhance our system in several ways. First, we aim to expand our dataset with more Hindi-English code-mixed tweets to expose the model to a wider range of expressions. Second, we'll refine our data preprocessing by translating Hindi-English utterances into plain English to reduce ambiguity. Additionally, we'll explore models beyond BERT, like LLAMA and GPT-2, known for text generation and question answering tasks. We'll also investigate specialized models like HingBERT and its family models for improved accuracy in Hindi-English code-mixed text analysis.

In essence, our future research focuses on dataset expansion, preprocessing improvements, and exploring diverse models to better understand emotions in multilingual conversations.

References

Thenmozhi D., Senthil Kumar B., Srinethe Sharavanan, and Aravindan Chandrabose. 2019. SSN_NLP at SemEval-2019 task 6: Offensive language identification in social media using traditional and deep machine learning approaches. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 739–744, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

DevlinJ, Chang M, Lee K, and Toutanova K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *Computing Research Repository*, arXiv:1810.04805.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Joosung Lee. 2022. The Emotion is Not One-hot Encoding: Learning with Grayscale Label for Emotion Recognition in Conversation. In *Proc. Interspeech 2022*, pages 141–145.

Himanshu Maheshwari and Vasudeva Varma. 2022. An ensemble approach to detect emotions at an essay level. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 276–279, Dublin, Ireland. Association for Computational Linguistics.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2:1–135.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances.

Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou. 2016. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):496–509.

Kushal Tatariya, Heather Lent, Johannes Bjerva, and Miryam de Lhoneux. 2024. Sociolinguistically informed interpretability: A case study on hinglish emotion classification.

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. *J. Am. Soc. Inf. Sci. Technol.*, 63(1):163–173.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Corpus creation and emotion prediction for Hindi-English code-mixed social media text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 128–135, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in Hindi-English code-mixed data: A transformer based approach. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 195–202, Online. Association for Computational Linguistics.

Yaoting Wang, Yuanchao Li, Paul Pu Liang, Louis-Philippe Morency, Peter Bell, and Catherine Lai. 2023. Cross-attention is not enough: Incongruity-aware dynamic hierarchical fusion for multimodal affect recognition.

KInIT at SemEval-2024 Task 8: Fine-tuned LLMs for Multilingual Machine-Generated Text Detection

Michal Spiegel[1,2] and **Dominik Macko**[1]
[1] Kempelen Institute of Intelligent Technologies
[2] Faculty of Informatics, Masaryk University
michal.spiegel@intern.kinit.sk, dominik.macko@kinit.sk

Abstract

SemEval-2024 Task 8 is focused on multigenerator, multidomain, and multilingual black-box machine-generated text detection. Such a detection is important for preventing a potential misuse of large language models (LLMs), the newest of which are very capable in generating multilingual human-like texts. We have coped with this task in multiple ways, utilizing language identification and parameter-efficient fine-tuning of smaller LLMs for text classification. We have further used the per-language classification-threshold calibration to uniquely combine fine-tuned models predictions with statistical detection metrics to improve generalization of the system detection performance. Our submitted method achieved competitive results, ranking at the fourth place, just under 1 percentage point behind the winner.

1 Introduction

Recent large language models (LLMs) are able to generate high-quality texts that are not easily detectable by human readers. A problem arises when such generated texts are misused for academic exams (Achiam et al., 2023), plagiarism (Wahle et al., 2022), disinformation spreading (Vykopal et al., 2023), etc. Therefore, it is crucial to develop automated means to detect machine-generated texts.

SemEval-2024 Task 8 (Wang et al., 2024) consists of three subtasks: A) binary human-written vs. machine-generated text classification, B) multi-way machine-generated text classification, and C) human-machine mixed text detection. In our work, we have focused on subtask A, especially its multilingual track. It covered 8 known languages for training (Arabic, Bulgarian, Chinese, English, German, Indonesian, Russian, Urdu), multiple domains (e.g., Wikipedia, news, abstracts), and multiple text generators (e.g., GPT-3, ChatGPT, BLOOMZ).

During our participation in the shared task, we have explored various alternatives. Our best submitted solution (illustrated in Figure 1) combines two fine-tuned LLMs (green-colored) with statistical detection (orange-colored) using a two-step majority voting (purple-colored) based ensemble method. Such a system achieved fourth place in the final leaderboard, with a performance of 95% in accuracy, within 1 percentage point range behind the winning system. We have published the source code for easier replication purposes[1]. We have used the statistical detection methods implemented in the recently published IMGTB framework[2] (Spiegel and Macko, 2023), which will be extended to also support all the fine-tunning options that we have used in this work.

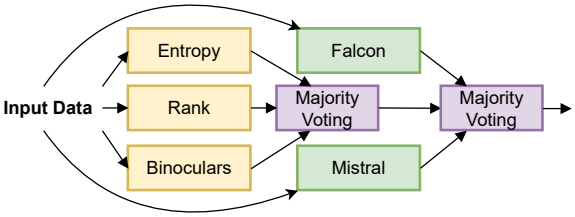


Figure 1: System components overview.

Our key contributions include:

- We have **proposed a unique way of combining the statistical and fine-tuned detection methods** using a two-way majority voting and a per-language threshold calibration.
- We have **proposed and compared three ensemble system alternatives** to cope with multilingual machine-generated text detection (additional two in the post-deadline study).
- We have **experienced a remarkably good performance of fine-tuned LLMs** of 7B parameters in this task.

[1] https://github.com/kinit-sk/semeval-2024-task-8-machine-text-detection
[2] https://github.com/kinit-sk/IMGTB

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 558–564

- We have proposed **the best-performing single-model system** called rMistral (Mistral-7B fine-tuned in a robust way – using both the train and dev sets and obfuscating 20% of the train data), achieving **0.97 AUC ROC** on the test data.

An interesting observation regarding our rMistral system is that although our per-language threshold calibration method would not bring the best accuracy on the test set (0.93), the threshold fixed to 1.0 (only predictions with a probability of 1, i.e. 100% confident, are marked as machine-generated) would win the competition (accuracy of 0.97). Nevertheless, we have noticed such a threshold performance only after the deadline and we considered the model being over-fitted (we would not submitted the results) which turned-out to be false.

2 Background and Methodology

For the machine-generated text detection task, three main groups of methods are nowadays used (Uchendu et al., 2023). The first one is a *stylometric detection*, which uses linguistic features (e.g., n-grams) to differentiate between human and machine writing styles (Fröhling and Zubiaga, 2021; Kumarage et al., 2023). The second group is a *statistical detection*, which uses statistical distribution based on a pre-trained language model (e.g., GPT2) to calculate various metrics (e.g., entropy) that can be used even without training (i.e., better generalization) to differentiate machine and human written texts (Mitchell et al., 2023; Hans et al., 2024). The last group is a *fine-tuned detection*, which further trains an already pre-trained language model for the detection task (Uchendu et al., 2020; Macko et al., 2023).

In the SemEval2024 Task 8 (Wang et al., 2024), we have focused on the multilingual track of Subtask A, which aimed at a binary classification to differentiate between human-written and machine-generated texts. The provided dataset (not allowing additional training data) contained the predefined splits of train, dev, and test sets. The train and dev sets officially contained 8 languages (3 languages in the dev set only), while unknown number of languages is contained in the test set.

Due to a multilingual nature of the data and our previous experience in multilingual machine-generated text detection (Macko et al., 2023), we wanted to try-out something new in this shared task. Our initial idea was to experiment with a per-language "mixture-of-experts", which would consist of multiple models, fine-tuned in a monolingual way per each official language in the train and dev sets. Since it was expected that surprise languages will be present in the test set, we would have used an additional multilingually fine-tuned model for other languages. However, we have started the experiments only few weeks before the deadline, which gave us little time to cope with the problems such as over-fitting and hyper-parameter optimisation (shown as severe towards the deadline).

Therefore, while training these per-language models, we also started to fine-tune the Falcon-7B model (Almazrouei et al., 2023) for the machine-generated text detection task, inspired by the winning system (Gagiano and Tian, 2023) of the recent ALTA 2023 shared task (although English monolingual). Since Falcon-7B is pre-trained on two languages only (English and French), we did not want to use it in a standalone way due to uncertain cross-lingual capability. Therefore, we have similarly fine-tuned the Mistral-7B model (Jiang et al., 2023), which is similarly sized generative model outperforming even some 13B parameters models in common benchmarks. We have not previously experimented with such a "big hammer" for the task; therefore, it was an interesting new experience for us. We have further combined these LLMs with statistical detectors to ensure better generalization of the system, which is described in the following sections.

3 System Overview

Our best system (see Figure 1) combines the predictions of two fine-tuned LLMs (Falcon-7B and Mistral-7B) with the selected statistical metrics (Entropy, Rank, Binoculars) by using **a two-step majority voting**. Firstly, a single majority-voted prediction results out of the three statistical metrics. Then, the final majority-voted prediction is a combination of the previous one with the Falcon and Mistral predictions.

Each prediction uses a separate classification-decision threshold, which is applied on prediction probabilities and statistical metrics. These **thresholds are calibrated in a per-language way**, meaning that separate thresholds are used for each language officially present in the train and dev sets, plus an additional threshold for unknown languages (i.e., not officially present in the train and dev sets). The thresholds are calibrated based on the machine-

System	Description
∗LLM2S3	The system described in this paper. It is an ensemble using two-step majority voting for predictions, consisting of 2 LLMs (Falcon-7B and Mistral-7B) fine-tuned using the train set only, 3 zero-shot statistical methods (Entropy, Rank, Binoculars) using Falcon-7B and Falcon-7B-Instruct for calculation of the metrics, utilizing language identification and per-language threshold calibration.
PLMoE	Our initial idea representing a per-language mixture of experts. It uses Electra-Large-Discriminator for English and XML-RoBERTa-Large for each of other languages officially present in the train and dev sets. Models for languages present in the dev set only are trained using the dev set. For unknown languages the Mistral-7B fine-tuned using the whole train set is used.
rLLM2S3	The same ensemble system as LLM2S3; however, the LLMs are fine-tuned using both the train and dev sets. Also, to increase the robustness of the system, we have obfuscated 20% of the train samples during fine-tuning, by using HomoglyphAttack and inserting zero-width-joiner character, inspired by our recent work (Macko et al., 2024).
rLLM2B-ES	The post-deadline ensemble system similar to rLLM2S3; however, the Llama-2-7B is used instead of Falcon-7B and Binoculars is used solely in the statistical part (instead of a combination of 3 methods). Moreover, the fine-tuning process used the early stopping mechanism to alleviate the over-fitting.
LLM2B1	The post-deadline ensemble system using the original LLM2S3 fine-tuned Falcon-7B and Mistral-7B models; however, classification thresholds are not calibrated, but only predictions with a probability of 1 (i.e., 100% confident) are marked as machine-generated. Such predictions are combined with Binoculars zero-shot prediction using the per-language threshold calibration.

Table 1: Description of system alternatives. The main system described in this paper is denoted by ∗. The last two alternatives were evaluated post-deadline.

class prediction probabilities and statistical metrics for samples in the train and dev sets combined. The calibration maximized the difference between true positive rate (TPR) and false positive rate (FPR) based on the ROC (receiver operating characteristic) curve. The texts with probabilities (or statistical metrics) outreaching the thresholds are considered machine-generated, otherwise they are considered human-written. The thresholds are saved and used for prediction of test samples.

Due to unknown languages in the test set and using the per-language threshold calibration, we have utilized the FastText[3] **language identification**. Since it is not fully accurate, we have used such language information only if the prediction probability was greater than 0.5, otherwise the language was handled as unknown.

As mentioned, the system includes **two fine-tuned LLMs**, namely Falcon-7B and Mistral-7B. For the fine-tuning process, we have used a parameter efficient fine-tuning (**PEFT**) technique called **QLoRA** (Dettmers et al., 2023) to minimize the computational costs of our system training.

To enhance the system performance generalization, we have integrated a **statistical part** of the system, which is based on the three statistical metrics, namely Entropy (Lavergne et al., 2008), Rank (Gehrmann et al., 2019), and recently proposed Binoculars (Hans et al., 2024). The statistical metrics are calculated using the Falcon-7B as a base model. Since Binoculars requires two models, it also uses Falcon-7B-Instruct (as a performer model).

Besides the described best submitted system, we have tried multiple system alternatives, which are briefly described in Table 1. In addition to those ensembles, we have evaluated single detectors, namely Falcon, Mistral, S5 (a combination of 5 statistical metrics – likelihood, entropy, rank, log-rank, and llm-deviation), and Binoculars. After the deadline, we have also finished fine-tuning of Llama-2-7B and retrained the detectors using the early stopping (patience of 5) to prevent overfitting. Also, when knowing the gold labels of the test set, we have evaluated various combinations of the trained detectors to see whether we have done the right decision for the submission.

4 Experimental Setup

For the experimental purpose, we have used the defaults splits of the provided dataset, namely the train and dev sets in the pre-deadline experiments, and the gold labels of the test set for the post-deadline evaluation of the pre-deadline system alternatives. The main system described in this paper uses only the train set in the training process; however, uses both the train and dev sets for the classification threshold calibration. Some of the

[3] https://pypi.org/project/fasttext-langdetect/

system alternatives used both the train and dev sets in the training process, as described in Table 1.

As the key evaluation metric in the shared task is **accuracy**, we have also used this metric for the preliminary system evaluation and selection of the alternative for submission. Since classification task is sensitive to the used classification threshold, we have also used **AUC ROC** (area under curve of the receiver operating characteristic) as a threshold independent metric, providing better information about the classification capability.

For the fine-tuning process, we have used the official baseline script[4], modified to export machine-class prediction probabilities in addition to the predictions. Since, it was not clear which version of the XLM-RoBERTa model was marked as a baseline in the multilingual track (with the known accuracy of 0.72), we have trained both the base (*XLM-R-B*) and large (*XLM-R-L*) versions. In addition, we have also included mDeBERTa-v3-base (*mDeBERTa*) model in our baselines, since it performed the best in our previous work (Macko et al., 2023).

To perform per-language models fine-tuning, we have used the source field of the train and dev data to select data only for the specific language. Other parameters of the fine-tuning process remained unchanged. The FastText language identification is used for a prediction, which uses the machine-class probability of the corresponding language-specific model.

The used QLoRA PEFT fine-tuning process used the binary cross entropy with logits for loss calculations and 4-bit quantization using BitsAndBytes[5]. The LoRA configuration[6] used an *alpha* of 16, a *dropout* of 0.1, *r* of 64, and the *task type* of sequence classification. Unlike the baseline fine-tuning, this version used half-precision training, gradient accumulation of 4 steps, and evaluation each 1,000 steps. Other parameters were the same.

Due to time constraints, we have not done any hyper-parameter optimization; thus, further improvements of the system are very likely possible.

5 Results

The experimental results are provided in Table 2. It must be noted that the results in the bottom part of the table are not part of the competition, since those experiments were performed after the submission deadline of the shared task. Also, the performance results using the test set were not known before the deadline; gold labels have been released only afterwards. Therefore, the design decisions could be made purely using the dev set.

Due to high accuracy and high AUC ROC metrics using the dev set, we considered *rFalcon* and *rMistral* over-fitted; therefore, we decided not to submit our *rLLM2S3* system. This turned-out to be a mistake, since it performed slightly better than the submitted *LLM2S3* on the test set. On the other hand, our suspicion of over-fitting *PLMoE* (due to the similar observations) turned-out to be valid, since it performed much worse using the test set. Therefore, it seems that per-language monolingually fine-tuned (i.e., lower amount of samples than multilingually fine-tuned) models require optimization of hyper-parameters to prevent over-fitting and to better generalize to unseen texts.

As an ablation study, we also provide the results for individual components of our system alternatives. As the results show, the ensembling into more complex systems of *LLM2S3* and *rLLM2S3* helped generalization of the classification performance. Individual methods would not outperform the submitted system.

5.1 Post-Deadline Study

In the post-deadline experiments (already knowing the gold labels of the test set for evaluation), we have finished Llama-2-7B model fine-tuning and retraining all three robust-version LLMs using the early stopping (to minimize the over-fitting). The results revealed that the *rLlama-2* model does not suffer by over-fitting as much. Based on the test set evaluation and by examining various combinations, the retrained *rLlama-2-ES* and *rMistral-ES* seemed like good candidates to combine with Binoculars (*rLLM2B-ES*), outperforming the winning system in the competition.

Early stopping helped a lot in boosting performance generalization (i.e., reducing over-fitting) of our per-language mixture-of-experts ensemble system (*PLMoE-ES*), achieving one of the highest AUC ROC using the test set. Nevertheless, in the accuracy as an official metric, it would not outperform the other system alternatives.

In addition, we have noticed that optimal thresholds for fine-tuned LLMs are often set to 1.0 by using purely the dev set samples machine-class

[4] https://github.com/mbzuai-nlp/SemEval2024-task8/blob/main/subtaskA/baseline/transformer_baseline.py

[5] https://pypi.org/project/bitsandbytes

[6] https://pypi.org/project/peft

	System	Accuracy		AUC ROC	
		Dev	Test	Dev	Test
Baselines	XLM-R-B	0.7158	0.7935	0.8262	0.9040
	XLM-R-L	0.7275	0.8841	0.8187	0.9063
	mDeBERTa	0.6968	0.8943	0.7952	0.9832
System Alternatives	∗LLM2S3•	0.9035	0.9501	N/A	N/A
	PLMoE•	0.9878	0.5819	0.9943	0.6268
	rLLM2S3•	0.9965	0.9560	N/A	N/A
Ablation Study	Falcon	0.8043	0.9102	0.8775	0.9492
	Mistral	0.8560	0.9027	0.9138	0.9579
	rFalcon	0.9905	0.8843	0.9991	0.9395
	rMistral	**0.9980**	0.9268	**0.9997**	0.9713
	S3•	0.7248	0.8328	N/A	N/A
	S5•	0.5880	0.4763	N/A	N/A
	Binoculars	0.5430	0.7979	0.6304	0.8777
	Binoculars•	0.6240	0.8434	0.6304	0.8777
Post-Deadline Study	PLMoE-ES•	0.9885	0.8417	0.9947	0.9635
	Llama-2	0.7335	0.7587	0.9342	0.7400
	rLlama-2	0.8903	0.8907	0.8416	0.9400
	rLlama-2-ES	0.9838	0.8805	0.9960	0.9108
	rFalcon-ES	0.9410	0.8672	0.9872	0.9503
	rMistral-ES	0.9863	0.9412	0.9984	**0.9834**
	rLLM2B-ES•	0.9915	0.9700	N/A	N/A
	LLM2B1	0.8668	**0.9708**	N/A	N/A
	rMistral1	0.9975	0.9675	**0.9997**	0.9713

Table 2: Detection performance evaluated using the dev (pre-deadline) and test (post-deadline) splits separately. The main system described in this paper is denoted by ∗, the systems using the per-language threshold calibration are denoted by •, systems using fixed threshold of 1.0 are denoted by "1". "-ES" denotes using of early-stopping mechanism to prevent over-fitting. "N/A" denotes not available values due to prediction-based majority voting (i.e., no probabilities to calculate AUC ROC). The gray color denotes unrepresentative performance values due to training on the dev set.

probabilities. Therefore, we have fixed the thresholds to 1.0 for the *LLM2B1* system (containing only models we have trained before the deadline), meaning that the machine-class predictions of the LLMs are used only when having 100% confidence (otherwise considered human-written). Such predictions, when combined with Binoculars, achieved even higher performance using the test set data (0.9708). Thus, we had such a system trained before the deadline; however, we have not noticed such a threshold bringing the best performance in time. Moreover, when looking at the accuracy for the dev set, we do not see why we would select such a system for the submission. It can be just a coincidence that it performs so well using the test set data. Further experiments are required to examine this phenomenon using independent out-of-distribution data.

Also, even when our *rLLM2B-ES* system alternative or the *rMistral1* single-model system would win the competition, we are now not sure that we would be confident enough (about not being overfitted) to submit it as the final system without evaluation on the external dataset. Thus, we have submitted best what we could at the time.

5.2 Per-language Analysis

For a deeper insight of the proposed system (*LLM2S3*) performance, we have performed an analysis per each language identified in the test set. The results are provided in Figure 2. Interesting is that it achieved the highest accuracy for the Italian surprise language (*it*). Lower accuracy is evident for German and Arabic languages, although present in the dev set. It must be noted that this version of the system was not trained using the dev set, only the classification threshold calibration used such data. Therefore, the robust versions of system alternatives are expected to provide higher performance especially in those languages.

6 Conclusion

To cope with the problem of multilingual, multidomain, and multigenerator machine-generated text detection, we have proposed an ensemble sys-

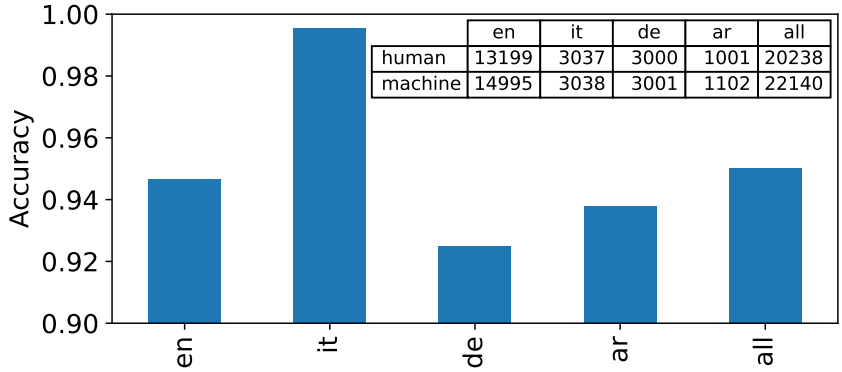


Figure 2: Per-language test-set performance (*it* is a surprise language, *de* and *ar* are in the dev set only). Axis scale for Accuracy is shown from 0.9 to 1.0. The per-class samples counts are provided in the top-right table.

tem using 2 LLMs (Falcon-7B and Mistral-7B) fine-tuned for the binary sequence classification task. We have further combined the predictions with the statistical metrics of Entropy, Rank, and Binoculars using a two-stage majority voting. The classification thresholds in our system have been calibrated in a per-language manner, for which we have utilized the FastText language identification. A combination of fine-tuned LLMs and statistical detection seems to be the right way to cope with generalization of the detection performance. Out of the evaluated single-model systems, Mistral-7B is the best candidate for fine-tuning, which by itself can bring a remarkable classification performance. Further improvements of the system could be easily achievable by hyper-parameters optimization, which we have not done in the submitted system due to lack of time.

Acknowledgements

This work was partially supported by the projects funded by the European Union under the Horizon Europe: *AI-CODE*, a project funded by the European Union under the Horizon Europe, GA No. 101135437, and *VIGILANT*, GA No. 101073921. Part of the research results was obtained using the computational resources procured in the national project *National competence centre for high performance computing* (project code: 311070AKF2) funded by European Regional Development Fund, EU Structural Funds Informatization of Society, Operational Program Integrated Infrastructure.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: An open large language model with state-of-the-art performance.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized llms.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. *PeerJ Computer Science*, 7:e443.

Rinaldo Gagiano and Lin Tian. 2023. A prompt in the right direction: Prompt based classification of machine-generated text detection. In *Proceedings of ALTA*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023. Stylometric detection of ai-generated text in twitter timelines. *arXiv preprint arXiv:2303.03697*.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse - Volume 377*, PAN'08, page 27–31, Aachen, DEU. CEUR-WS.org.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Dominik Macko, Robert Moro, Adaku Uchendu, Ivan Srba, Jason Samuel Lucas, Michiharu Yamashita, Nafis Irtiza Tripto, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2024. Authorship obfuscation in multilingual machine-generated text detection. *arXiv preprint arXiv:2401.07867*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Michal Spiegel and Dominik Macko. 2023. IMGTB: A framework for machine-generated text detection benchmarking. *arXiv preprint arXiv:2311.12574*.

Adaku Uchendu, Thai Le, and Dongwon Lee. 2023. Attribution and obfuscation of neural text authorship: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 25(1):1–18.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. 2023. Disinformation capabilities of large language models. *arXiv preprint arXiv:2311.08838*.

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. How large language models are transforming machine-paraphrase plagiarism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. SemEval-2024 Task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

A Computational Resources

For experiments regarding model fine-tuning and inference processes, we have used $1\times$ NVIDIA GeForce RTX 3090 24GB GPU and $1\times$ A100 40GB GPU, cumulatively consuming around 10,000 GPU-core hours. For combining the results and analysis, we have used Jupyter Lab running on 4 CPU cores, without the GPU acceleration.

Sharif-MGTD at SemEval-2024 Task 8: A Transformer-Based Approach to Detect Machine Generated Text

Seyedeh Fatemeh Ebrahimi ♣, **Karim Akhavan Azari**♣, **Amirmasoud Iravani**★
Arian Qazvini♢, **Pouya Sadeghi**†, **Zeinab Sadat Taghavi**♣, **Hossein Sameti**♣
Ferdowsi University of Mashhad, Mashhad, Iran★
Amirkabir University of Technology, Tehran, Iran♢
University of Tehran, Tehran, Iran†
Sharif University of Technology, Tehran, Iran♣

{sfati.ebrahimi, karim.akhavan, zeinabtaghavi, sameti}@sharif.edu
a.iravani@mail.um.ac.ir
a.qazvini@aut.ac.ir
pouya.sadeghi@ut.ac.ir

Abstract

Detecting Machine-Generated Text (MGT) has emerged as a significant area of study within Natural Language Processing. While language models generate text, they often leave discernible traces, which can be scrutinized using either traditional feature-based methods or more advanced neural language models. In this research, we explore the effectiveness of fine-tuning a RoBERTa-base transformer, a powerful neural architecture, to address MGT detection as a binary classification task. Focusing specifically on Subtask A (Monolingual - English) within the SemEval-2024 competition framework[1], our proposed system achieves an accuracy of 78.9% on the test dataset, positioning us at 57th among participants. Our study addresses this challenge while considering the limited hardware resources, resulting in a system that excels at identifying human-written texts but encounters challenges in accurately discerning MGTs.

1 Introduction

Recent advancements in large language models (LLMs) have endowed them with an impressive capability to generate written text that closely resembles human writing (Adelani et al., 2019; Radford et al., 2019). However, this technological progress brings along significant challenges, as the proliferation MGT poses various threats in digital environments. MGTs have been implicated in spreading misinformation in online reviews, eroding public trust in political or commercial campaigns, and even facilitating academic fraud (Crothers et al., 2022; Song et al., 2015; Tang et al., 2023). The identification of MGT remains a pressing concern, as distinguishing between human-written and machine-generated content is often challenging for humans. Consequently, there is a growing imperative to develop automatic systems capable of discerning MGT (Mitchell et al., 2023). In this study, we address this challenge within the English language context using the dataset provided by Wang et al. (2023).

As highlighted in Wang et al. (2024b) overview paper on the task, recent approaches to MGT detection predominantly employ binary classification methods. Existing literature highlights the superior performance of transformer-based methods over alternative approaches Wang et al. (2024a). However, a significant challenge in utilizing these models lies in the requirement for GPU hardware and computational resources. Our study aims to address this challenge within the constraints of limited hardware capacity. Keeping this in mind, we propose a system that leverages fine-tuning of the RoBERTa transformer model (Liu et al., 2019) to automatically classify input text as either human-written or machine-generated. Our system architecture involves augmenting the RoBERTa-base model with a Classifier Head. The Embeddings component facilitates contextual understanding of texts, while the Encoder component processes input texts in parallel, and the Classifier Head performs binary classification by linearly outputting a single value.

Our proposed system achieves an accuracy of 78.9% on the test data, surpassing the average results provided by the task's baseline and ranking 57th among 140 participants. The area under the

[1] https://semeval.github.io/SemEval2024/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 565–572

ROC curve (AUC) metric is measured at 0.69. While the ROC curve analysis demonstrates our model's capability to classify substantial portions of positive cases, its proximity to the diagonal line indicates room for further improvement. Notably, our primary challenge stemmed from computational constraints, which limited our ability to implement larger token sizes or batch sizes. Further discussions reveal that our system encounters difficulties in accurately detecting MGTs. To facilitate reproducibility and further research in this area, the code for our system is available on GitHub[2].

2 Background

2.1 Dataset Overview

SemEval-2024 Task 8 (Wang et al., 2024b) comprises three subtasks, with our investigation centering on Subtask A: binary classification of human-written versus MGT. Specifically, we concentrated our efforts on analyzing English monolingual data, as outlined dataset is provided by Wang et al. (2023).

Subtask A encompasses a dataset consisting of 119,757 training examples and 5,000 development examples, all presented in JSON format. Each data instance includes the following attributes:

- *id*: An identifier number for the example.
- *label*: A binary label indicating whether the text is human-written (0) or machine-generated (1).
- *text*: The actual textual content.
- *model*: The AI machine responsible for generating the text.
- *source*: The web domain from which the text originates.

2.2 Related Work

MGT detection is feasible through both traditional feature-based methods and neural language models. Fröhling and Zubiaga (2021) and Nguyen-Son et al. (2018) discussed how feature-based methods leverage statistical techniques. These methods primarily utilize frequency features such as TF-IDF, linguistic cues, and text style (Fröhling and Zubiaga, 2021). However, feature-based methods have limitations, as different samplings in language models can lead to varied generated outputs (Holtzman et al., 2019). In contrast, methods that harness neural language models, particularly those employing transformer models, have shown high effectiveness (Crothers et al., 2022). Neural language model methods often involve zero-shot classification or fine-tuning pre-trained language models (Sadasivan et al., 2023). Grover by Zellers et al. (2019), RankGen by Krishna et al. (2022), and DetectGPT (Mitchell et al., 2023) are prominent examples of zero-shot methods. However, these methods may be misleading at times and exhibit limited performance in out-of-domain tasks (Crothers et al., 2022; Wang et al., 2023).

Bakhtin et al. (2019) demonstrated outstanding performance in MGT detection by harnessing bidirectional transformers. Additionally, Solaiman et al. (2019) highlight that the zero-shot methods often fall short compared to a simple TF-IDF baseline when detecting texts from diverse domains. He argues that bidirectional transformers offer significant advantages for MGT detection, advocating for the fine-tuning of these models as a superior alternative to zero-shot methods. In this regard, Rodriguez et al. (2022) observed a significant enhancement in performance of cross-domain MGT detection by fine-tuning the RoBERTa detector.

Jawahar et al. (2020) conducted a comprehensive survey of various approaches to developing MGT detectors. Their findings suggest that fine-tuning the RoBERTa detector consistently delivers robust performance across diverse MGT detection tasks, surpassing the efficacy of traditional machine learning models and neural networks. Additionally, Crothers et al. (2022) reported a notable trend towards the increased utilization of bidirectional transformer architectures, particularly RoBERTa, in MGT detection tasks. Lastly, Wang et al. (2024a) conducted a comprehensive benchmark of supervised methods on M4 dataset. Their findings revealed that transformer models such as RoBERTa and XLM-R exhibited superior performance across all tests, respectively achieving 99.26% and 96.31% accuracy in MGT binary classification.

While this review does not provide a comprehensive examination of all aspects of

[2] https://github.com/Sharif-SLPL/Sharif-MGTD

MGT detection, prior research underscores the prevalence of transformer-base methods, like RoBERTa and XLM-R, in comparison to alternative approaches, especially in supervised tasks. Moreover, the superiority of RoBERTa over other models is evident. A significant challenge for studies utilizing pre-trained transformer models lies in the necessity for robust GPU hardware and computational resources.

3 System Overview

This section presents an overview of our system's architecture, highlighting implementation details and challenges. Drawing on the preceding works discussed above, which showed the efficacy of fine-tuning RoBERTa models, our system aims to attain peak performance in MGT detection while optimizing configurations for limited hardware resources.

The decision to employ the transformer architecture for detecting synthetic texts is motivated by its capacity to capture intricate dependencies within textual data. This choice seems logical considering that such texts often exhibit semantic features that can be harnessed for fact-checking, cohesion, coherence, and other properties that may unveil their origin (Raj et al. (2020). In contrast to traditional architectures, the transformer model overcomes the constraints of fixed window sizes or sequential processing, enabling it to utilize contextual information from the entire input sequence. Additionally, the self-attention mechanism empowers the model to selectively focus on pertinent segments of the input, rendering it highly effective for tasks necessitating long-range dependencies and contextual comprehension.

As for RoBERTa, it is specifically chosen for its extensive training duration, broader dataset coverage, ability to handle longer sequences, and focus on Natural Language Understanding tasks, making it more suitable than other BERT-based models. Additionally, a wealth of research, such as the recent study of Wang et al. (2024a), has further highlighted the inherent potential of RoBERTa for this specific task.

3.1 Core Algorithms and System Architecture

At the core of our system lies the concept of binary classification, distinguishing input texts as either machine-generated or human-written through fine-tuning a pre-trained RoBERTa transformer (Liu et al., 2019). Our system architecture entails augmenting the RoBERTa-base model with a Classifier Head. The RoBERTa model's Embeddings component incorporates a 768-dimensional embedding matrix, alongside position and token type embeddings, enhancing contextual understanding. The Encoding component features a 12-layer RoBERTaEncoder, each layer employing a multi-head self-attention mechanism. This facilitates simultaneous attention to different parts of the input text, crucial for analyzing textual similarities. Intermediate sub-layers utilize a fully connected feed-forward network with GELU activation, followed by an output sub-layer for feature transformation and normalization.

The Classifier Head, integrated into the Encoder for sequence classification, comprises a linear layer with 768 input features and a dropout layer to mitigate over-fitting. The final output is generated through an additional linear layer with a solitary output neuron, making it conducive to binary classification tasks. In essence, the primary model processes input data, with the Classifier Head making predictions. When viewed as a regression task, the Classifier produces a linear output tailored for a singular class, providing a probabilistic value. Implementation of the system is facilitated using PyTorch, incorporating specific parameters such as the AdamW optimizer (Radford and Narasimhan, 2018) and the CrossEntropyLoss function (Hui and Belkin, 2020). AdamW, renowned for training deep neural networks, integrates weight decay to mitigate over-fitting. The Cross Entropy Loss function, commonly employed in multi-class classification scenarios, combines softmax activation with negative log-likelihood loss. The training process involves iterating through the entire dataset for two epochs, with early stopping mechanisms in place to terminate training at the optimal point.

3.2 System Challenges

While larger machine-generated documents often exhibit more discernible patterns and clues, such as incoherence or repetition, they also entail substantial computational costs. Our primary challenge lay in efficiently processing these large documents using cost-effective computing systems. To mitigate this challenge, we explored strategies such as reducing token size and batch size. However, these adjustments necessitate trade-offs, potentially leading to reduced accuracy or

increased processing time.

Our system was trained using a token size of 512, but optimal performance could potentially be achieved with larger token sizes, such as 1024 or 2048, given sufficient computing resources.

4 Experimental Setup

4.1 Dataset

Table 1 presents detailed statistics on the dataset used for each class.

Class/Split	Train	Test	Development
Human-Written Text	57075	6276	2500
Machine-Generated Text	50706	5700	2500

Table 1: Dataset Statistics

As shown in Table 1, nearly 90% of the dataset is dedicated to training, while the remainder is used for evaluation. To enhance model performance, we utilized the entire development dataset for model selection, compensating for the scarcity of training data.

4.2 Pre-processing and Hyper-Parameter Tuning

Input texts are tokenized using the RoBERTa tokenizer before processing, both during training and inference. Our hyper-parameter tuning process involved a comprehensive exploration across various parameter ranges. Specifically, we conducted experiments with learning rates ranging from 0.0001 to 0.00004, dropout rates spanning from 0.1 to 0.3, batch sizes varying between 4 and 16, and token sizes ranging from 64 to 1024. Through experimentation and analysis, we determined the optimal hyper-parameter settings, which are as follows: a learning rate of 0.00004, a dropout rate of 0.1, a token size of 512, a batch size of 10, and a weight decay of 0.01. Further details are given in Appendix A.

As illustrated in Appendix A, the number of training instances is correlated with the input token size and may influence the model accuracy. Given the length of input texts, a suitable token size is essential to capture all tokens adequately. However, computational costs associated with larger token sizes present a significant challenge during model training. Consequently, we selected 512 as the optimal token size. Truncation was employed during tokenization to accommodate the chosen token size, ensuring efficient model training without compromising data representativeness.

4.3 Training Procedure

For training the model, we utilized the Task dataset Wang et al. (2023), which underwent preprocessing by tokenizing the text into sub-word units and padding sequences to a fixed length. CrossEntropyLoss was employed as the loss function. The implementation also involved the AdamW optimizer, known for its effectiveness in training deep neural networks and its incorporation of weight decay to address over-fitting. The Adam optimizer was utilized with a learning rate of 4e-05. During training, the loss was monitored on a held-out validation set, and early stopping was applied to prevent over-fitting. Early stopping was implemented with the condition that the training loss reached a specific threshold (0.35 in this case), typically occurring around the third epoch. Therefore, if there was no improvement in the validation loss for a certain number of epochs, training was halted to prevent over-fitting of the model.

4.4 Evaluation Measures

The evaluation of our model involves calculating its accuracy in predicting whether a text is human-written or machine-generated. Accuracy, a fundamental metric in classification tasks, assesses the overall correctness of predictions and is calculated as:

$$Accuracy = \frac{ni}{N} \times 100 \quad (1)$$

where n_i represents the number of correctly classified instances, and N is the total number of instances.

5 Results

Using the official accuracy metric of SemEval-2024 Task 8 (Wang et al., 2024b), our system achieved the following accuracy scores on different data splits:

Language /Split	Devset	Testset
English	74.8%	78.9%

Table 2: Accuracy Metric

A direct comparison of our results with prior works is challenging due to the unique nature of

our research. To the best of researchers' knowledge, the most comprehensive benchmark on supervised MGT detection is presented by Wang et al. (2024a) using the M4 dataset and employing RoBERTa, XLM-R, GLTR-LR, GLTRSVM, Stylistic-SVM, and NELA-SVM. However, our primary objective was to determine strategies for addressing limited hardware resources as discussed in Appendix A.

As a contribution to this field, through repeated experiments, we identified that among hyperparameters, token size plays a slightly more significant role in model accuracy. While the system's accuracy is influenced by increasing the token size, drawing meaningful scientific conclusions necessitates further controlled experiments. Additionally, the expansion of token size is restricted by hardware limitations, requiring a detailed investigation with robust computational resources like GPU or TPU. Considering the constraints of Google Colab's[3] Free runtimes, we opted for a token size of 512 as a balance between hardware limitations and time constraints. Consequently, based on the official accuracy metric of SemEval2024 Task 8 (Wang et al., 2024b), our system achieved the following accuracy scores on various data splits:

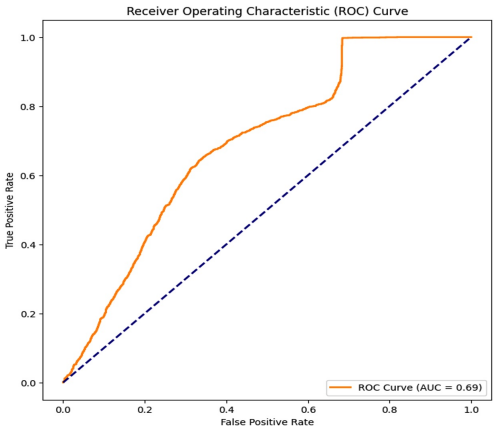


Figure 1: The ROC Curve Plot

The evaluation of our model also included analysis of the Area Under the Curve (AUC), a crucial metric that reflects the discriminative power of a binary classification model. Our fine-tuned RoBERTa model demonstrated an AUC of 0.69, suggesting its ability to effectively distinguish between positive and negative instances. Figure 1 illustrates the Receiver Operating Characteristic

[3] https://colab.research.google.com

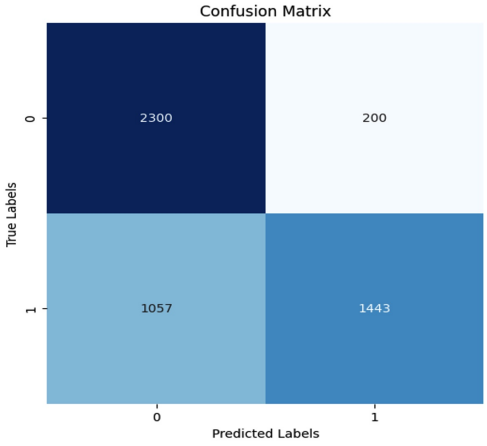


Figure 2: The Confusion Matrix Plot

(ROC) Curve, depicting the model's capability to accurately classify a significant proportion of positive cases. However, the proximity of the curve to the diagonal line suggests opportunities for further enhancement.

Interestingly, analysis of the confusion matrix, as depicted in Figure 2, revealed notable patterns in our model's classification tendencies. While our system effectively identified human-written documents with low False Positives, it exhibited difficulties in correctly identifying MGTs. This observation suggests potential areas for refinement, particularly in enhancing the model's ability to detect subtle cues and characteristics unique to machine-generated content.

Overall, our study contributes to the ongoing efforts in the field of NLP by showcasing the effectiveness of fine-tuned transformer models, particularly RoBERTa, in MGT detection tasks. Moving forward, future research directions could explore novel approaches to mitigate computational costs and further improve the performance of MGT detection systems, ultimately advancing the capabilities of NLU models in real-world applications.

6 Conclusion

In summary, our study focused on fine-tuning a RoBERTa-base transformer model for binary classification, specifically in distinguishing human-written from MGT. While our system showed promise in identifying human-written text, it faced challenges with accurately classifying machine-generated content. As discussed in Appendices A and B, we recommend exploring larger token sizes to improve model performance, albeit with

awareness of computational costs. Additionally, we advocate for the development of low-cost algorithms capable of efficient processing across hardware platforms. Our findings contribute to advancing MGT detection, with implications for combating misinformation and enhancing cybersecurity in the digital age.

Acknowledgments

We appreciate the Speech and Language Processing Laboratory at Sharif University of Technology[4] for providing us with this opportunity for collaborative work.

References

David Ifeoluwa Adelani, Hao Thi Mai, Fuming Fang, Huy Hoang Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *International Conference on Advanced Information Networking and Applications*.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *ArXiv*, abs/1906.03351.

Evan Crothers, Nathalie Japkowicz, and Herna L. Viktor. 2022. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *ArXiv*, abs/1904.09751.

Like Hui and Mikhail Belkin. 2020. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *ArXiv*, abs/2006.07322.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. In *International Conference on Computational Linguistics*.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. *ArXiv*, abs/2205.09726.

[4] https://github.com/Sharif-SLPL

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*.

Hoang-Quoc Nguyen-Son, Ngoc-Dung T. Tieu, Huy Hoang Nguyen, Junichi Yamagishi, and Isao Echizen. 2018. Identifying computer-generated text using statistical analysis.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Mayank Raj, Ajay Jaiswal, Rohit R.R, Ankita Gupta, Sudeep Kumar Sahoo, Vertika Srivastava, and Yeon Hyang Kim. 2020. Solomon at SemEval-2020 task 11: Ensemble architecture for fine-tuned propaganda detection in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1802–1807, Barcelona (online). International Committee for Computational Linguistics.

Juan Diego Rodriguez, Todd Hay, David Gros, Zain Shamsi, and R. Srinivasan. 2022. Cross-domain detection of gpt-2-generated technical text. In *North American Chapter of the Association for Computational Linguistics*.

Vinu Sankar Sadasivan, Aounon Kumar, S. Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *ArXiv*, abs/2303.11156.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *ArXiv*, abs/1908.09203.

Jonghyuk Song, Sangho Lee, and Jong Kim. 2015. Crowdtarget: Target-based detection of crowdturfing in online social networks. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *ArXiv*, abs/2303.07205.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar

Habash, Iryna Gurevych, and Preslav Nakov. 2024a. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. *ArXiv*, abs/2402.11175.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *ArXiv*, abs/2305.14902.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *ArXiv*, abs/1905.12616.

A Hyper-Parameter Tuning

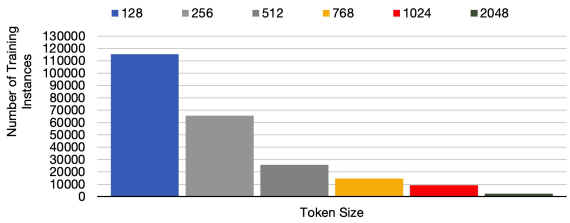


Figure 3: Number of Training Instances by Token Size

To determine the appropriate settings for hyper-parameters, we utilized Google Colab's free GPU runtime. Free Colab users have access to GPU and TPU runtimes without charge for a maximum of 12 hours. The GPU runtime includes an NVIDIA Tesla K80 with 12GB of VRAM. [Date: 5 Dec 2023]. We were unable to use premium runtime accounts due to financial issues arising from Iran sanctions. Therefore, we couldn't change our model's token size to larger than 512 due to the 12-hour time limit in free Colab. To understand the impact of increasing token size, we aimed to experiment on a local laptop GPU.

During the experiments aimed at finding the proper token size, we encountered the "CUDA error: device-side assert triggered" frequently, which was resolved by restarting the session. Our experiments were conducted using an RTX 2060 mobile with 6 GB of VRAM. Throughout all experiments, we maintained fixed parameters, including Number of Epochs = 3, Train Split = 0.7, and Learning Rate = 4e-05. Increasing the Max Length from 512 to 1024 in this experimental setup resulted in an improvement in Test Accuracy by at least 2%. However, this enhancement came at the cost of a nearly 15-fold decrease in training speed, making it challenging to implement on limited hardware. Additionally, this requires plenty of controlled experiments by researchers to shed light on finding the proper hyper-parameters.

B Detect-GPT as a Zero-Shot Method

In our pursuit of effective MGT detection, we also experimented with Mitchell et al. (2023)Detect-GPT model, a zero-shot approach utilizing probability curvature analysis. Training the model resulted in an accuracy rate of 60%, and when applied to a test dataset of approximately 1500 samples, it achieved a remarkable accuracy of approximately 84%. We conducted a comprehensive analysis by implementing 10 perturbations for each dataset. To address data and mask filling tasks, we employed the T5 small model, leveraging its robust capabilities. Furthermore, to accurately assess the log likelihood, we utilized the GPT-2 model, ensuring precise calculations and reliable results.This method surpassed alternative text detection methodologies, demonstrating superior accuracy and reliability in identifying MGT. Notably, the inclusion of threshold configuration added granularity to the experiment, enabling fine-tuning of detection sensitivity across varying threshold settings.

IRIT-Berger-Levrault at SemEval-2024: How Sensitive Sentence Embeddings are to Hallucinations?

Nihed Bendahman [♢♠], **Karen Pinel-Sauvagnat** [♢],
Gilles Hubert [♢], **Mokhtar Boumedyen Billami** [♠]
[♢]Université Paul Sabatier, IRIT, UMR 5505 CNRS, Toulouse, France
[♠]Berger-Levrault, 64 Rue Jean Rostand, 31670 Labège, France
{nihed.bendahman, karen.sauvagnat, gilles.hubert}@irit.fr
{nihed.bendahman, mb.billami}@berger-levrault.com

Abstract

This article presents our participation to Task 6 of SemEval-2024, named SHROOM (*a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes*), which aims at detecting hallucinations. We propose two types of approaches for the task: the first one is based on sentence embeddings and cosine similarity metric, and the second one uses LLMs (Large Language Model). We found that LLMs fail to improve the performance achieved by embedding generation models. The latter outperform the baseline provided by the organizers, and our best system achieves 78% accuracy.

1 Introduction

In recent years, with the emergence of the foundation models, the generation of hallucinated text has become an increasingly prominent and alarming issue. Despite the state-of-the-art performances achieved by the latest text generation models, such as GPT-4 (Achiam et al., 2023) or Llama 2 (Touvron et al., 2023), the problem of hallucinations remains open, making these models challenging to apply in real-world applications.

Hallucination is defined as a segment of text that appears fluent and natural but contains incoherent and inconsistent information compared to the provided context (Ji et al., 2023). The problem of hallucinations appears in several NLG tasks such as text summarization (Cao et al., 2021; Zhang et al., 2022) and machine translation (Xu et al., 2023). The shared-task Shroom[1] falls within the scope of these tasks.

The aim of Shroom is to identify samples containing hallucinations with regard to the provided context through a binary classification. The task is established in a post hoc setting, where models have already been trained to generate text based on the provided context. Three text generation tasks are considered: machine translation (MT), paraphrasing generation (PG), and definition modeling (DM). One can find in Table 1 a sample of data for the Machine Translation task. Participants should find the *label* of the *hypothesis* that was generated by the model, given the *target* or the *source*. For instance, here the aim is to determine if the hypothesis (*I've got the floor and the furniture.*) is a hallucination given the source (*J'ai poli le plancher et les meubles.)* or the target (*I polished up the floor and furniture.*). In this example, the hypothesis is labeled as hallucination (*label*) with regard to the assessments made by 3 annotators (*labels*).

Source : J'ai poli le plancher et les meubles.
Target : I polished up the floor and furniture.
Hypothesis : I've got the floor and the furniture.
Ref : Either
Labels : [Hallucination, Hallucination, Hallucination]
Label : Hallucination
p(hallucination) : 1.0

Table 1: Data sample

The binary classification can be performed in two different tracks: the model-aware and model-agnostic tracks. In the model-aware track, the checkpoint of the model that generated the hypothesis is provided and can be used in the classification system, which is not the case for the model-agnostic track. For more information on task 6 of SemEval-2024 Shroom as described by its organizers, we invite the reader to consult the paper by (Mickus et al., 2024).

In the literature, detecting hallucinations in sentences mainly relies on comparing segments of text. Among these approaches, one can cite those based on named entities (Nan et al., 2021): the idea is to determine if the generated text contains incon-

[1] https://helsinki-nlp.github.io/shroom/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 573–578

sistent entities compared to the source. Other approaches are based on question-answering methods, where the aim is to answer a set of questions and evaluate the difference between the two texts (Wang et al., 2020). However, data provided for the Shroom task are of a particular nature as it consists of very short texts, composed of one or two concise sentences. This characteristic limits the use of several detection methods, such as those based on named entities, since they are very rare in these texts. The same goes for question-answering methods. This observation has led us to turn to simpler methods, involving capturing the semantics of sentences in a general way.

In this paper, we present the two lines of approaches we investigated for the task:

- The first one is based on embedding models. We solve the hallucination classification task using the cosine similarity metric between sentence embeddings of the reference and hypothesis (see Section 4.1),
- The second one relies on Large Language models with specified prompts to classify the generated text that contains hallucinations. We use Llama 2 and Mistral using 2 different prompts inspired by SelfCheckGPT (Manakul et al., 2023).

Our sentence embedding approaches outperformed our LLM ones. The former have proven to be very relevant given the shortness of the sentences and the low risk of losing information. Our best performing system obtained the accuracy of 78% in both tracks which puts us in position 22/48 in the model-agnostic track and 22/45 in the model-aware track.

2 Related Work

Approaches in the literature can be classified depending on the hallucinatory content to detect (Ji et al., 2023). Some works focus on the detection and comparison of existing entities between the context and the generated hypothesis, such as (Nan et al., 2021). They are based on the assumption that human brain is sensitive to different types of information, such as named entities and proper nouns when reading, and mistakes concerning named entities are striking to human users (Ji et al., 2023). (Feng et al., 2023) go further by evaluating facts (entities and relations). Another line of works focuses on the use of question-answering as an indicator to identify hallucinations (Wang et al., 2020; Scialom et al., 2021), or the use of text entailment, which consists in determining whether the generated text is a hallucination if it cannot be entailed by the source (Falke et al., 2019).

Other approaches focused on the classification of hallucination types (whether they are intrinsic or extrinsic (Maynez et al., 2020)), or factual or non-factual (Cao et al., 2021).

Large Language Models have also been used to determine whether the generated text contains hallucinatory content. The aim of these methods is to set up prompts to compare the sources and the hypotheses (Manakul et al., 2023; Chern et al., 2023). Other methods make specific prompts to ask the LLM to "think" and judge whether a given text contains hallucinations, justifying its answer by producing a chain-of-thought (CoT) explanation (Friel and Sanyal, 2023).

In this paper, we explore a simpler method that consists in calculating the embeddings of the hypothesis and the reference, and computing their semantic similarity using the cosine similarity metric. Contextual embeddings have been successfully used in various NLP tasks, such as sentiment analysis (Carrasco and Dias, 2023) or topic modeling (Schneider, 2023). Our underlying idea here is to see how sensitive the semantic similarity is to the hallucinated content in sentences and to what extent the cosine similarity metric reflects this sensibility.

3 Data Description

The organizers of Shroom provided data from 3 different NLG tasks : Machine Translation, Paraphrasing Generation, Definition Modeling. Each task is divided into two tracks: model-aware track and model-agnostic track. We were provided 5 datasets in the development phase : train-aware, train-agnostic, trial, dev-aware and dev-agnostic, containing 30000, 30000, 80, 501 and 499 samples respectively and 2 different test datasets in the evaluation phase: test model-aware and test model-agnostic, containing each 1500 samples.

For each sample, the model-generated hypothesis was annotated by 3 (trial dataset) or 5 different annotators (dev and test datasets) (*labels* in Table 1). Annotators were asked to assess whether the generated hypothesis was consistent with the reference and to provide a label {hallucination, not-hallucination}. At the end of the annotation process, the most preponderant label is chosen as the

final label (*label* in Table 1), with an assigned probability corresponding to the proportion of annotators who considered this specific datapoint to be an hallucination (*p(hallucination)* in Table 1).

4 System Overview

As the training dataset provided by organizers is not labeled, we decided to experiment unsupervised approaches, either using sentence embeddings or LLMs.

4.1 Embedding-Based Approach

We first generate the contextual embeddings of the reference and the hypothesis. For the paraphrasing generation task, we consider the source as the reference, while for the other two tasks, the target is taken as the reference. Next, we calculate the cosine similarity between these two embeddings. If this similarity does not exceed a predefined threshold, we assign the label "hallucination". We evaluated various embedding models namely Sentence-T5 XL (Ni et al., 2021), a specialized variant of the T5 model designed specifically to generate representations of sentences; BGE-base; BGE-large (Xiao et al., 2023); E5-base; E5-large and SF E5 (Wang et al., 2022). We compared their performances to determine their effectiveness in our task using different cosine similarity thresholds (see Section 5.2). This comparison enabled us to select the most appropriate model with the cosine threshold that maximizes the classification accuracy.

Participants were also asked to estimate a probability of the predicted labels. To estimate this probability, we apply an empirical rule. Let t be our threshold, $cossim$ the value of the cosine similarity, l our predicted label et $p(l)$ the probability of hallucination. Algorithm 1 details the rules we applied.

The idea behind this rule is that the further we are from the cosine similarity threshold, the more certain we are that the hypothesis generated by the language model is a hallucination.

4.2 LLM-Based Approach

We tested two LLMs, Llama-2-13b (Touvron et al., 2023) and Mistral-7b (Jiang et al., 2023), with 2 different prompts inspired by SelfCheckGPT (Manakul et al., 2023). This enabled us to make a direct comparison with the baseline system given by the organizers, which is based on a variant of Mistral fine-tuned with the same first prompt we used.

Algorithm 1 Hallucination Probability Estimation

if $cossim \geq t$ **then**
  $l \leftarrow Not\ Hallucination$
  if $cossim \leq t + \epsilon$ **then**
    $p(l) \leftarrow 0.33$
  else
    $p(l) \leftarrow 0.0$
  end if
else
  $l \leftarrow Hallucination$
  if $cossim \geq t - \epsilon$ **then**
    $p(l) \leftarrow 0.66$
  else
    $p(l) \leftarrow 1.0$
  end if
end if

As our method uses only the generated hypotheses and the references to detect hallucinations, we used a single model for both model-aware and model-agnostic tracks.

5 Experiments and Results

5.1 Experimental Setup

For all the models used, we retrieved the checkpoints from the HuggingFace website [2].

- For the embedding-based approach, we experimentally fix ϵ to 0.05 (see Algorithm 1) using the dev-set. t is also fixed experimentally. Experiments conducted to determine its value are detailed in section 5.2.

- For the LLMs approach, we use the Langchain framework [3] to set up the prompts and query the LLMs. Table 2 describes the two prompts we used. Prompt 1 is directly taken from SelfCheckGPT's system (Manakul et al., 2023) which serves as Baseline. The idea of the work of (Manakul et al., 2023) is to ask the LLM an explicit and simple question. They show that with this kind of prompts, LLMs better understand the task they are asked to perform. With regard to prompt 2, we wanted to experiment whether introducing the concept of "hallucination" in the prompt and specifying its definition helps the LLM better classify.

[2] hhttps://huggingface.co/
[3] https://www.langchain.com/

Prompt 1 (Manakul et al., 2023)	Context:{} Sentence:{} Is the Sentence supported by the Context above? Answer using ONLY yes or no:
Prompt 2	Context:{} Sentence:{} Is the Sentence a hallucination (which means it contains inconsistent or incoherent information) compared to the Context above? Answer using ONLY yes or no:

Table 2: Prompts used to perform the binary classification.

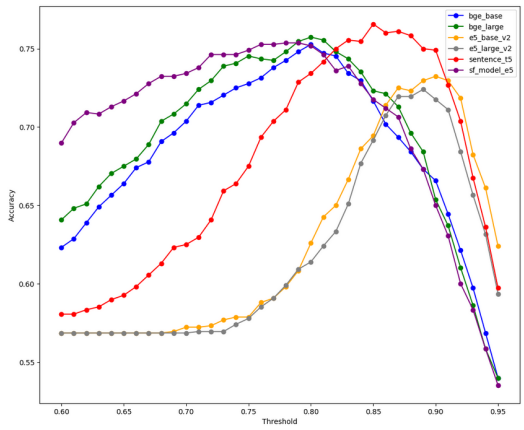


Figure 1: Variation of the threshold of the cosine similarity metric maximizing the accuracy of the classification system as a function of the models over the Trial and Dev datasets.

5.2 Preliminary Experiments

In this section, we describe the experiments made to determine the threshold of the cosine similarity metric that maximizes classification accuracy (t in Algorithm 1). We defined an interval ranging from 0.6 to 0.95. Then, for each value of the interval by step of 0.01, we performed the classification and calculated its accuracy. This experimentation was conducted on the 3 labelled datasets, namely trial, dev-aware and dev-agnostic. The graph in Figure 1 shows the evolution of accuracy as a function of the cosine similarity threshold used to define hallucination. We can see that the models behave in a similar way: accuracy rises progressively with the threshold values used, reaching a peak around the [0.78, 0.9] interval. We can also see that for the models for which we used two variants as BGE-base, BGE-large as well as E5-base and E5-large, the behavior of the variants is almost identical with a few small differences. Table 3 reports the selected threshold for each model applied on the test dataset in the evaluation phase.

5.3 Results

Two evaluation metrics are used for the task: the accuracy of the classification and the Spearman correlation of the system's output probabilities with the proportion of the annotators marking the item as hallucinated. The official ranking was made on the basis of the accuracy, with a tie-breaker between systems having obtained the same score using the Spearman correlation. As we did not provide classification probabilities for the LLMs approach, we only report them for the embedding-based approach.

Table 3 shows the results we obtained with the different embedding models submitted. We can see that all the models exceed the baseline on the two metrics used, with the best performance coming from the Sentence-T5 model. Given that the baseline consists in the use of an LLM with a prompt, we can say that the embedding models used with the right threshold distinguish fairly well between hallucinated and non-hallucinated hypotheses compared to LLM with the used prompts. Since organizers published official results and released the test sets, we re-ran our experiments varying t, threshold used with the cosine metric. The results, not reported here, are consistent with those of the trial and dev collections. This leads us to believe that our approach of threshold selection is robust.

Table 4 shows the results obtained with the LLMs we used. We can see that they do not perform as well as the embedding models, and do not exceed the baseline. Regarding the prompts we used, no conclusion can be drawn for the moment. Further experiments are required.

With the scores obtained by the sentence-T5 model, we were ranked 22/45 and 22/48 in the model-aware and model-agnostic tracks respectively. It is worth noting that the first half of the ranking is extremely tight. It often takes 4 decimal digits to separate the accuracy of the various participants.

6 Conclusion

In this paper, we summarize our participation to task 6 of the SemEval-2024 evaluation campaign:

Model	Value of t for Aw	M-Aw	SC-Aw	Value of t for Ag	M-Ag	SC-Ag
Baseline	/	0.745	0.487	/	0.696	0.402
Best system	/	0.812	0.699	/	0.847	0.769
Worst system	/	0.483	-0.06	/	0.460	0.133
BGE-Base	0.77	0.750	0.552	0.79	0.754	0.563
BGE-Large	0.77	0.766	0.569	0.78	0.766	0.581
E5-Base	0.87	0.742	0.494	0.90	0.748	0.531
E5-Large	0.87	0.754	0.510	0.89	0.751	0.525
Sentence T5 XL	0.86	**0.781**	**0.601**	0.85	**0.782**	**0.636**
SF E5	0.75	0.758	0.523	0.79	0.762	0.540

Table 3: Results obtained (accuracy (M) and Spearman correlation (SC)) with each embedding model using the selected threshold, in comparison to the Baseline, best and worst submitted systems. Results are reported for the model-aware (Aw) and model-agnostic (Ag) tracks.

Model	M-Aw	M-Ag
Llama-2-13b-chat Prompt 1	0.618	0.557
Llama-2-13b-chat Prompt 2	0.555	0.536
Mistral-7b-instruct Prompt 1	0.627	0.519
Mistral-7b-instruct Prompt 2	0.676	0.618

Table 4: Results obtained (accuracy M) for each LLM with the two prompts used. Results are reported for the model-aware (Aw) and model-agnostic (Ag) tracks.

Shroom. We presented two different approaches: one based on the use of embedding models with a cosine similarity threshold to perform the binary classification, and the other based on LLM using simple prompts to detect hallucinatory content. We showed that on the data used for the task, embedding generation models perform better than LLMs. In future work, we will explore this approach a little further, by fine-tuning the models used for instance.

Acknowledgements

This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011014740 made by GENCI.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2021. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. *arXiv preprint arXiv:2109.09784*.

Paulo Carrasco and Sandra Dias. 2023. Exploring natural language processing and sentence embeddings for sentiment analysis of online restaurant reviews. *Atas da 23ª Conferência da Associação Portuguesa de Sistemas de Informação*.

Ethan Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai - a tool augmented framework for multi-task and multi-domain scenarios. *ArXiv*, abs/2307.13528.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Annual Meeting of the Association for Computational Linguistics*.

Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. 2023. Factkb: Generalizable factuality evaluation using language models enhanced with factual knowledge. *ArXiv*, abs/2305.08281.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *ArXiv*, abs/2310.18344.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Potsawee Manakul, Adian Liusie, and Mark John Francis Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *ArXiv*, abs/2303.08896.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. *ArXiv*, abs/2005.00661.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cícero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. Entity-level factual consistency of abstractive text summarization. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Matthew Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *ArXiv*, abs/2108.08877.

Johannes Schneider. 2023. Efficient and flexible topic modeling using pretrained embeddings and bag of sentences. *ArXiv*, abs/2302.03106.

Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. 2021. Questeval: Summarization asks for fact-based evaluation. In *Conference on Empirical Methods in Natural Language Processing*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *ArXiv*, abs/2004.04228.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv*, abs/2212.03533.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *ArXiv*, abs/2309.07597.

Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J Martindale, and Marine Carpuat. 2023. Understanding and detecting hallucinations in neural machine translation via model introspection. *Transactions of the Association for Computational Linguistics*, 11:546–564.

Haopeng Zhang, Semih Yavuz, Wojciech Kryscinski, Kazuma Hashimoto, and Yingbo Zhou. 2022. Improving the faithfulness of abstractive summarization via entity coverage control. *arXiv preprint arXiv:2207.02263*.

CYUT at SemEval-2024 Task 7: A Numerals Augmentation and Feature Enhancement Approach to Numeral Reading Comprehension

Tsz-Yeung Lau
Department of CSIE
Chaoyang University of Technology
Taichung, Taiwan
s10927116@gm.cyut.edu.tw

Shih-Hung Wu†
Department of CSIE
Chaoyang University of Technology
Taichung, Taiwan
shwu@cyut.edu.tw

Abstract

This study explores Task 2 in NumEval-2024, which is SemEval-2024(Semantic Evaluation) Task 7 , focusing on the Reading Comprehension of Numerals in Text (Chinese). The dataset utilized in this study is the Numeral-related Question Answering Dataset (NQuAD), and the model employed is BERT. The data undergoes preprocessing, incorporating Numerals Augmentation and Feature Enhancement to numerical entities before model training. Additionally, fine-tuning will also be applied. The result was an accuracy rate of 77.09%, representing a 7.14% improvement compared to the initial NQuAD processing model, referred to as the Numeracy-Enhanced Model (NEMo).

1 Introduction

Numeric information holds a crucial significance within narratives across various domains, including medicine, engineering, and finance. (Chen et al., 2021) Numerals presented in tables (Ibrahim et al., 2019) and the content (Lamm et al., 2018) of a document have garnered considerable attention from researchers. Machine-based numeral comprehension stands out as an emerging research area, still in its nascent stages. NumEval-2024 Task 2(SemEval-2024 Task 7) focus on reading comprehension of the numerals in text, models are required to identify the correct numerical value from four given options, based on a provided news article. The dataset utilized for this task is NQuAD (Chen et al., 2021), which is in Chinese.

The initial model devised for addressing this task is referred to as the Numeracy-Enhanced Model (NEMo), which achieves an accuracy of 69.95%.(Chen et al., 2021) In this study, a pre-trained BERT model will be employed to undertake the task, with the objective of enhancing accuracy through data preprocessing, Numeral Augmentation, Feature Enhancement, and fine-tuning. [1]

[1]†Contact Author

The remaining sections of this study are structured as follows: In the second section, we delve into a discussion of the data and its preprocessing. The data undergoes denoising through the removal of special symbols.

Moving on to the third section, we thoroughly examine the methodology employed, the results obtained, and the subsequent discussion. The data undergoes further denoising through stop word removal. Additionally, numeral augmentation and feature enhancement are introduced. Numeral augmentation reduces dependence on specific numerical values, while feature enhancement aims to compel the model to pay attention to numerals. As a result, our model achieves an accuracy of 77.09% on the NQuAD dataset.

The concluding section presents a summary of the insights gained, along with considerations for future research endeavors. Although the model is performing well, there are opportunities for further enhancement. We will delve into these matters in the Error Analysis and Discussion section.

2 Data Preparation

2.1 Data Source

The dataset employed in this study is the Numeral-related Question Answering Dataset (NQuAD). This dataset poses greater challenges compared to numeral-related questions in other datasets. All data were sourced from news articles spanning the period from June 22, 2013, to June 20, 2018, encompassing a total of 75,448 Chinese news articles. Notably, 59.74% of news headlines include at least one numeral, while numerals are present in 99.80% of news contents. The dataset comprises 43,787 news articles and 71,998 questions.(Chen et al., 2021)

The NQuAD dataset encompasses six columns: "news_article","question_stem","answer_options", "ans","target_num" and "sentences_containing_the

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 579–585

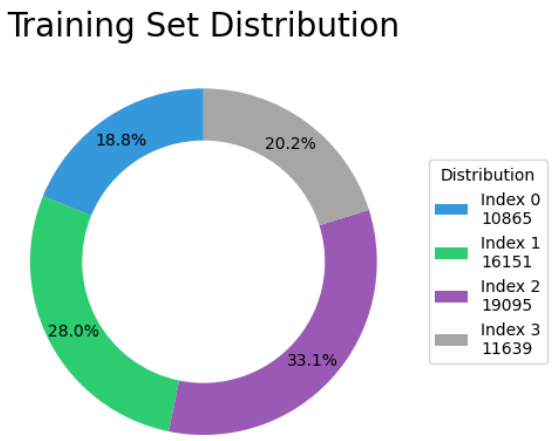


Figure 1: Training Set Distribution

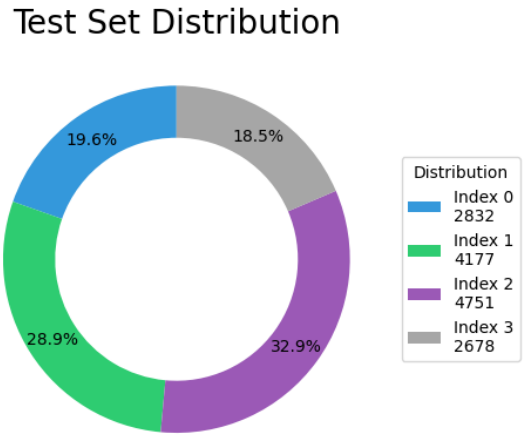


Figure 2: Test Set Distribution

_numeral_in_answer_options". The "news_article" column contains the content of the article, while the "question_stem" column represents the questions posed. The "answer_options" column consists of a list of four answer choices, with the "ans" column indicating the index of the correct answer within the "answer_options." The "target_num" column contains the content of the correct answer, and the "sentences_containing_the_numeral_in_answer_options" (scao) is a list of sentences in the article that include the correct answer.

NQuAD provides both a training set and a test set. The training set comprises 57,750 samples, while the test set includes 14,438 samples, maintaining an approximate 8:2 ratio. The distribution of the four categories varies within the training set, encompassing 10,865 instances where the correct answer index is 0, 16,151 instances for index 1, 19,095 instances for index 2, and 11,639 instances for index 3. In the test set, there are 2,832 instances with the correct answer index at 0, 4,177 instances at index 1, 4,751 instances at index 2, and 2,678 instances at index 3. The data depicted in Figure 1 and Figure 2 reveals that approximately 20% is attributed to both index 0 and index 3, while around 30% is associated with both index 1 and index 2.

2.2 Data Preprocessing

The dataset used in this study, NQuAD, provides the "scao" column, which constitutes a list of sentences within the article containing the correct answers. A new column, denoted as "article", is formed by combining the contents of the "scao", "question_stem" and "answer_options" columns. The "article" column serves as the input for the model. Additionally, the "ans" column, representing the index of the correct answer within the "answer_options" column, is employed to generate a new column named "label". The "label" column functions as the output for the model.

After creating the new input column "article", the content of the "article" column undergoes preprocessing. Special symbols such as "[# &" are removed, taking care not to eliminate symbols that may affect the semantics, such as "+ - . % $". Subsequently, HTML tags are removed, along with excess whitespaces within sentences. Finally, commas in numbers are removed, for instance, in cases like "1,000", to facilitate subsequent batch processing.

Please be aware that pre-processing will be applied to both the training set and the test set.

3 Method

In this study, the pretrained BERT model served as the baseline. To enhance model performance, two steps were implemented. First, BERT with stop word removal (BERT-SWR) was introduced. This step contributes to improved model performance through Dimensionality Reduction, Noise Reduction, and Enhanced Generalization. Building upon BERT-SWR, additional measures were taken, including numerals augmentation and feature enhancement(BERT-NAFE). Numerals in the text were replaced with a special symbol and the corresponding answer index. This substitution aimed to eliminate the model's consideration of the meaningless numerical values in the article. The use of the special symbol, represented as cashtag, compelled the model to prioritize attention to

the numerals within the text. This additional step further increased the model's accuracy. Please be aware that prior to evaluation, fine-tuning was applied to each model.

3.1 BERT

A pretrained BERT model was used as the baseline model. The complete designation for BERT is Bidirectional Encoder Representations from Transformers, a bidirectional unsupervised language representation model based on transformers, initially introduced by Google. It undergoes predominantly pre-training through the application of both the Masked Language Model (MLM) and Next Sentence Prediction (NSP) techniques. In contrast to word2vector (Pennington et al., 2014) and GloVe (Mikolov et al., 2013), which operate without considering context, BERT exhibits the capability to leverage contextual information during inference. This contextual understanding contributes to its superior performance across diverse tasks (Devlin et al., 2019).

3.2 Stop Word Removal

The preprocessing technique known as removing stop words in natural language processing (NLP) is intended to exclude commonplace yet generally uninformative words, such as "和", "你", "而是", etc., from the textual content. By eliminating these words, low-level information is excluded from the text, enabling a heightened emphasis on crucial information. This step is undertaken with the aim of enhancing model performance.

In this study, a tokenizer named jieba (Sun, 2020) was employed to segment the sentences. A stop-word list sourced from Sichuan University (goto456, 2019) was utilized to determine phrases that should be eliminated. The Python OpenCC package is employed to perform the translation from simplified Chinese to traditional Chinese for the stop-word list. For example, let's examine a phrase: "而是前一世代的iPhone 6s." Subsequent to the processing, the term "而是" is eliminated, resulting in the refined expression "前一世代的iPhone 6s."

3.3 NA and FE

3.3.1 Numerals Augmentation

Reducing the model's excessive reliance on specific numerical values and addressing numerical ambiguity are pivotal objectives in Numeral Augmentation for tasks related to Numeral Reading Comprehension.

article:
年增**22.09**% ,
精華 1565 董事昨 **23** 日通過去年盈餘分配案 ,
擬每股配發現金股利**27**元 , 營益率**34.82**% ,
精華去年EPS 35.26元寫新高 ·
擬每股配息___元 ,
22.09 , 23 , 27 , 34.82

answer_options: ['22.09', '23', '27', '34.82']

processed article:
年增**0**% ,
精華 1565 董事昨 **1** 日通過去年盈餘分配案 ,
擬每股配發現金股利**2**元 , 營益率**3**% ,
精華去年EPS 35.26元寫新高 ·
擬每股配息___元 ,
0 , 1 , 2 , 3

Figure 3: artical after NAFE pre-processing.

In this study, each answer's numeral will be replaced by its corresponding index, ranging from 0 to 3. By addressing these aspects, Numeral Augmentation contributes to enhancing the model's robustness and interpretative capabilities when confronted with diverse numerical contexts in reading comprehension tasks. For example, as illustrated in Figure 3, the numerical values in the answer options were replaced with their corresponding indices. Upon examining the processed article, the numbers 0, 1, 2, and 3 were underscored, indicating the replacements for the actual answers in the article.

3.3.2 Feature Enhancing

In this study, Feature Enhancement has been applied to numerical values. Specifically, the digits representing the answers in the text are augmented with cashtag both preceding and following them. Due to the attention mechanism of the model, there is a likelihood that attention may concentrate in the vicinity of these markers, thereby directing the model to place increased emphasis on processing the content surrounding the markers. This, in turn, enhances the model's capability for recognizing numerical. For instance, as demonstrated in Figure 3, ashtag were added both before and after the replaced answer. This step involves referencing the Tokenization Tricks presented in (Jiang et al., 2020).

3.4 Fine-tuning

Fine-tuning refers to the process in deep learning where a pre-trained model is utilized and further trained to adapt to specific tasks or domains. In this study, models based on BERT have undergone fine-tuning to enhance their ability to recognize numerical content in articles. The scrutinized parameters are delineated in Table 2. Further elaboration on these details will be provided in the Experimental Setup section.

3.5 BERT-NAFE

The optimal model in this study is BERT-NAFE. Details will be discussed in this section. Commencing with the data, symbols were eliminated during the data pre-processing, resulting in a refined dataset suitable for model training and evaluation. Expanding upon the foundational BERT model, the removal of stop words was implemented, resulting in a significant 3% increase in accuracy. Additionally, Feature Enhancement was employed, followed by Numerals Augmentation. During Numerals Augmentation, numerals were substituted with their index in the answer list, while being omitted from the article text. Cashtag were inserted before and after the substituted numeral as part of Feature Enhancement. This procedural refinement led to a notable 6% boost in accuracy and was shown to be the most effective in enhancing overall performance.

4 Experimental setup

4.1 Environment

The experimental setup is adaptable to both Google Colab and local environments. For local setup, a minimum of 6GB of GPU memory is necessary for model fine-tuning. The versions of each tool employed in the experiment will be detailed in the accompanying Table 1.

Parameters	Values
python	3.9.18
tensorflow	2.10
cudatoolkit	11.2
cudnn	8.1.0
ktrain	0.40.0

Table 1: Tools Versions Employed in the Experiment.

Parameters	Values
BERT Model	base
Batch size	3
Max length	250
Max learning rate	2e-5
epochs	2

Table 2: Parameters Value of Model Training.

4.2 Hyperparameter

For the fine-tuning parameters, a batch size of 3 and maximum sequence length of 250 were utilized in order to reduce computational expenses during fine-tuning. A max learning rate of 2e-5 was adopted based on hyperparameter tuning performed using the Ktrain learner. This approach was taken to identify the optimal learning rate, as visualized in Figure 4.

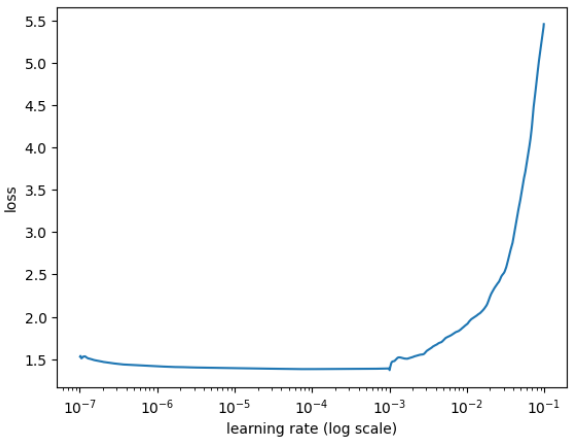


Figure 4: The loss vs. learning rate chart for finding best learning rate in our training processing experiment.

An assessment of validation performance across epochs indicated that achieving satisfactory model performance could be accomplished within just 2 epochs, as depicted in Figure 5. Additional epochs did not meaningfully improve results and instead prolonged training without benefit. Therefore, 2 epochs were deemed adequate for the model to learn from the data effectively.

Regarding the batch size, we conducted a search across values of 3, 4, 8, 16, and 32, all of which exhibited comparable performance. Consequently, we opted to utilize a batch size of 3 for our experiments.

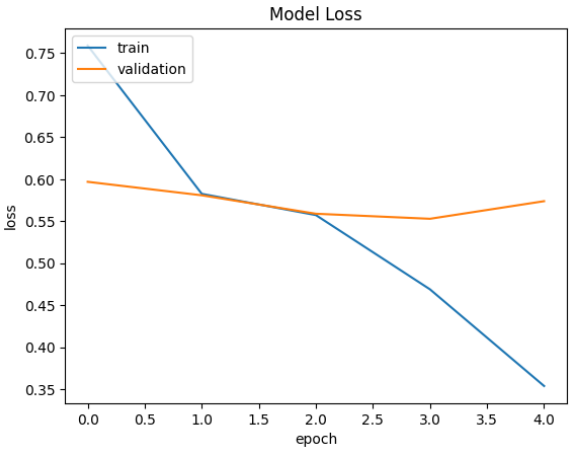


Figure 5: The loss vs. epochs chart during training.

5 Result

Table 3 illustrates the performance of each model developed in this study, comparing them with the initial NEMo model, which was created by the subsequent study (Chen et al., 2021), as indicated by the underlined model in Table 3. The BERT model exhibited an accuracy of 67.95%, a 2% decrease compared to the NEMo model. Upon removal of stop words, the BERT-SWR achieved an accuracy of 70.83%, presenting a marginal improvement of less than 1% compared to the NEMo model. In the case of the BERT-NAFE model, an accuracy of 77.09% was attained, reflecting a 7.14% increase compared to the NEMo model. The Jupyter Notebook employed in this investigation has been made publicly available on GitHub. For further reference, please consult the Appendix.

Model	Accuracy
Initial Model	
NEMo(Chen et al., 2021)	69.95%
Our Model	
BERT	67.95%
BERT-SWR	70.83%
BERT-NAFE	77.09%

Table 3: Comparison of model performance. (All models were fine-tuned except for NEMo.)

Therefore, it can be asserted that the removal of stop words, Numerals Augmentation, and Feature Enhancement are effective in enhancing accuracy, surpassing the performance of the existing model.

6 Discussion and Error Analysis

6.1 Discussion

The baseline model, BERT, achieved an accuracy of 67.95% after fine-tuning, closely approaching the initial model NEMo. Upon the removal of stop words from the text, there was a notable improvement of approximately 3% in accuracy. Considering dimensionality reduction, stop words, characterized as high-frequency terms, pervade most texts. Their exclusion significantly diminishes the dimensionality of the feature space, thereby reducing computational complexity. Consequently, this eases the requirements on both model training and inference processes.

The elimination of stop words contributes to noise reduction. These words often lack essential information, and retaining them may introduce interference, hindering the model's ability to concentrate on learning crucial information. Their removal enables the model to focus its attention and learning capacity on genuinely meaningful vocabulary. Additionally, excluding stop words during model training enhances the comprehension and generalization of the text. Stop words frequently serve as grammatical connectors without conveying specific semantic information. Their removal allows the model to more effectively concentrate on learning the actual relationships between words, unburdened by the intricacies of entire sentence structures.

In addition to BERT-SWR, Numerals Augmentation and Feature Enhancement were incorporated to further elevate the accuracy to 77.09%. Two reasons account for this enhancement. To address the model's inclination to overly rely on particular numerical values encountered during training, Numeral Augmentation aims to counteract this fixation. This is achieved by introducing variations in the representation of numbers, guiding the model to develop a less rigid fixation on specific instances and fostering a more generalized understanding of numerical information.

Another central focus of the augmentation process is the mitigation of numerical ambiguity. Numerals within a given context may have multiple interpretations or ambiguous meanings. To alleviate this ambiguity, the augmentation process exposes the model to diverse representations of numbers. This exposure aids the model in adapting to different contexts and facilitates the disambiguation of numeral meanings based on the surrounding textual

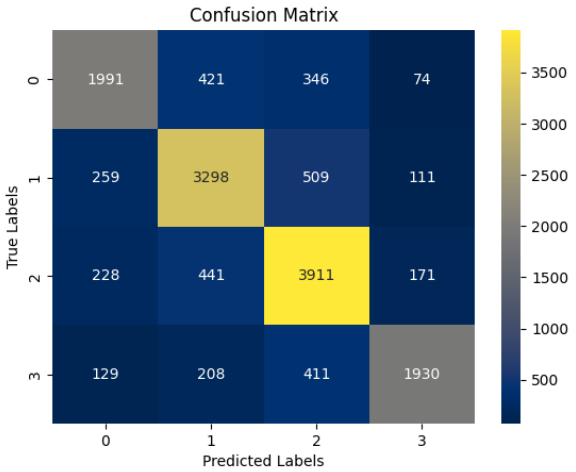


Figure 6: Confusion Matrix of model BERT-NAFE.

information.

After computing the current precision and recall from Figure 6 , it was observed that the system tends to produce class 2 outputs at a rate of 35.9%, surpassing the training set's rate of 33.1%. Conversely, the system's output proportion for class 3 is notably low, standing at only 15.8%. Perhaps adjusting the training set's distribution to a balanced 1:1:1:1 ratio could enhance the system's performance. It is worth noting that the numerical labels 0, 1, 2, and 3 used in the Numerals Augmentation step lack intrinsic significance and merely denote positions. For future work, adjusting the distribution ratio should be a straightforward task.

Various symbols and triple symbols for feature enhancement were also experimented with. In the case of different symbols, comparable performance was observed. However, when employing triple symbols, there was a slight decrease in accuracy, with approximately 74%, representing a 3% decline.

article:
每股虧損**0.88**元 , 至少有**2**位數的成長 ,
稅後虧損**5.06**億元 ,毛利率為**7.6%** ,
燿華：Q___獲利有很大改善，Q3至少二位數成長 ,
0.88 , **2** , **5.06** , **7.6**

question_stem:
燿華：Q___獲利有很大改善，Q3至少二位數成長
answer_options: ['0.88', '2', '5.06', '7.6']
target_num: 2

Figure 7: Example of Bad Sample

6.2 Error Analysis

In certain instances among the samples, the "scao" columns do not encompass the answers to the questions, requiring the model to predict the answer through estimation. For example, as illustrated in Figure 7, the question pertains to improving profit, yet the content in the "scao" column does not include this information.

To address this issue, the "news_article" can be employed as a substitute for the "scao" column. However, this substitution would result in an increase in token length, leading to a rise in training costs, encompassing GPU memory usage, and extending model training time.

7 Conclusions

This study has achieved significant performance improvements through fine-tuning the BERT model and optimizing text processing methods. Firstly, during the fine-tuning process, we observed a notable increase in model accuracy by removing stop words from the text, particularly high-frequency vocabulary. The exclusion of stop words not only reduced the dimensionality of the feature space, lowering computational complexity, but also contributed to noise reduction, enabling the model to better focus on meaningful vocabulary.

Secondly, the introduction of Numerals Augmentation and Feature Enhancement further elevated the model's accuracy. Numerals Augmentation, by introducing variations in the representation of numbers, assisted the model in overcoming overreliance on specific numerical values, fostering a more generalized understanding of numeric information. Additionally, Feature Enhancement strategies helped alleviate numerical ambiguity, allowing the model to adapt to different contexts and accurately interpret numeric meanings.

In summary, our research findings suggest that, building upon the BERT model, fine-tuning and text processing optimizations can effectively enhance performance in numeral reading comprehension.

Acknowledgements

This study was partially supported by the National Science and Technology Council under the grant number NSTC 112-2221-E-324-014.

References

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Nquad: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2925–2929, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

goto456. 2019. scu-stopwords.

Yusra Ibrahim, Mirek Riedewald, Gerhard Weikum, and Demetrios Zeinalipour-Yazti. 2019. Bridging quantities in tables and text. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1010–1021.

Mike Tian-Jian Jiang, Yi-Kun Chen, and Shih-Hung Wu. 2020. Cyut at the ntcir-15 finnum-2 task: Tokenization and fine-tuning techniques for numeral attachment in financial tweets. In *The 15th NTCIR Conference Evaluation of Information Access Technologies*, pages 92–96. NII Testbeds and Community for Information access Research.

Matthew Lamm, Arun Chaganty, Christopher D. Manning, Dan Jurafsky, and Percy Liang. 2018. Textual analogy parsing: What's shared and what's compared among analogous facts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 82–92, Brussels, Belgium. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Andy Sun. 2020. jieba.

Appendix

https://github.com/anson70242/BERT-NAFE

UniBuc at SemEval-2024 Task 2: Tailored Prompting with Solar for Clinical NLI

Marius Micluța-Câmpeanu[♣,♡,*], **Claudiu Creangă**[♣,♡,*]
Ana-Maria Bucur[♣,♡], **Ana Sabina Uban**[♠,♡], **Liviu P. Dinu**[♠,♡]
[♠] Faculty of Mathematics and Computer Science
[♣]Interdisciplinary School of Doctoral Studies, [♡]HLT Research Center
University of Bucharest, Romania
marius.micluta-campeanu@unibuc.ro, claudiu.creanga@s.unibuc.ro
ana-maria.bucur@drd.unibuc.ro, {auban, ldinu}@fmi.unibuc.ro

Abstract

This paper describes the approach of the UniBuc team in tackling the SemEval 2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. We used SOLAR Instruct, without any fine-tuning, while focusing on input manipulation and tailored prompting. By customizing prompts for individual CTR sections, in both zero-shot and few-shots settings, we managed to achieve a consistency score of 0.72, ranking 14th in the leaderboard. Our thorough error analysis revealed that our model has a tendency to take shortcuts and rely on simple heuristics, especially when dealing with semantic-preserving changes.

1 Introduction

Clinical trials are prospective studies that aim to compare the effectiveness of an intervention against a control group in clinical patients (Friedman et al., 2015). ClinicalTrials.gov[1] hosts more than 480,000 clinical trials, making it challenging to analyze and extract information from them manually. Natural language inference has emerged as a valuable tool for interpreting evidence from clinical trials (Jullien et al., 2023a).

The second task of SemEval 2024 focuses on improving the understanding of clinical trial data through the second edition of NLI4CT (Natural Language Inference for Clinical Trials) (Jullien et al., 2024). This challenge is specifically designed to test the natural language inference capabilities of large language models (LLMs) and their ability to understand clinical text. The data used for the challenge was carefully annotated by clinical domain experts, and semantic interventions were performed to evaluate the safety and robustness of the models.

We employed LLMs, achieving the best results with SOLAR-Instruct, and focused on two key strategies:

- **Targeted Summarization**: Summarizing both CTRs and the hypothesis (retaining only the first 'trial' sentence) aided the model's focus on essential information.
- **Tailored Prompting**: We used both zero-shot and two-shot prompting, tailoring prompts to individual CTR sections for optimal results.

We were surprised to find that causal models significantly outperform masked language models on this type of task. This is probably because the task requires reasoning capabilities that BERT-based models do not have. Our model's biggest challenges were with numerical reasoning and rephrasing (discussed in Section 6), but despite those, we still secured the 14th place (out of 32) in the leaderboard. We make our code publicly available on GitHub[2].

2 Related Work

Recent work on clinical trial analysis includes detecting contradictions in medical publications (Makhervaks et al., 2023), automating eligibility assessment with LLMs (Wang et al., 2023; Datta et al., 2024), and assessing model hallucinations and reasoning capabilities in healthcare settings (Pal et al., 2023; Feng et al., 2023). The previous SemEval NLI4CT task (Jullien et al., 2023b) included a similar entailment task, with most submissions leveraging pre-trained language models. A small minority of approaches used ontologies and rule-based algorithms.

Few of the language model-based approaches include preprocessing of the data prior to using it as input to the models. In our approach for the current

[*]These authors contributed equally to this work.
[1]https://clinicaltrials.gov/
[2]https://github.com/ClaudiuCreanga/sem-eval-2024-task-2

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 586–595

task, we also attempt solutions based on both discriminative and generative language models, and in addition perform preprocessing of the input clinical trial data before feeding it to the models, such as summarization. While most of the best performing systems in last year's task employ some in-domain pre-training on medical data, we find that, from the models we experimented with, general LLMs perform as well as or better than medical ones. This could be explained by their larger size and instruction tuning techniques, but also because of the recent advances in general LLMs. However, we do not perform an exhaustive comparison of the two kinds of models (we use few medical LLMs in our experiments), so a definitive conclusion can not be drawn on this comparison.

The NLI4CT dataset (Jullien et al., 2023a) is a unique benchmark dataset for Natural Language Inference (NLI) in the clinical domain that contains data from Clinical Trial Reports (CTRs). In contrast, the MedNLI (Romanov and Shivade, 2018) dataset is another benchmark dataset for NLI in the clinical domain, but it only contains clinical notes. To ensure that NLI models are robust and safe, the organizers of the NLI4CT task perform semantic-preserving and semantic-altering interventions of the hypotheses. According to Jullien et al. (2023a), NLI models for clinical trials require not only biomedical reasoning but also numerical reasoning capabilities, as CTRs contain a large amount of quantitative information. In this regard, we conduct experiments using SOLAR 10B (Kim et al., 2023), which was trained on question-answer pairs from the mathematical domain to enhance its mathematical capabilities.

3 Data and Task Description

The data used for this task is comprised of 1,000 breast cancer CTRs collected from ClinicalTrials.gov with 24,000 entailment relations annotated by clinical domain experts (Jullien et al., 2023a). Each CTR is comprised of 4 sections: eligibility criteria, intervention, results and adverse events.

Each sample from the NLI4CT dataset consists of the CTR premise (one of the 4 sections of the CTR), a statement, and an entailment label (Entailment or Contradiction). The premise can refer to only one CTR in the *Single* type instance, or to a primary and a secondary trial in the *Comparison* type. The purpose of the current task is to evaluate the consistency of models and their ability to perform faithful reasoning (Jullien et al., 2024). For this purpose, different semantic altering (Contradiction Rephrasing and Numerical Contradiction) or semantic preserving interventions (Paraphrase, Numerical Paraphrase and Definition) have been conducted on the evaluation data. The NLI4CT dataset consists of 1,700 entailment relations in the training split, 200 in dev, and 5,500 in the test split.

To assess the performance of the models in the shared task, two metrics have been proposed: **Faithfulness** and **Consistency** (Jullien et al., 2024), besides F1-score. Faithfulness measures the model's ability to adjust its predictions accurately after a semantic-altering intervention. Consistency measures the capacity of the system to predict the same label for both the original and contrast statements, in the case of semantic-preserving interventions.

4 Methodology

In this section, we present the different approaches used to predict the entailment relations.

4.1 Pre-trained Masked Language Models

Our first approach for the task of entailment relations prediction is using pre-trained models. Previous research has shown that domain-specific pretraining is beneficial for biomedical tasks (Gu et al., 2022; Romanov and Shivade, 2018).

We use pre-trained models, such as PubMedBERT (Gu et al., 2021), XLM-RoBERTa (Conneau et al., 2020), Deberta V3 Large (He et al., 2021), and fine-tune them on the training data. The fine-tuning process is done in 2 steps, as suggested in (Sun et al., 2020). Firstly, we stack a fully connected layer on top of the pre-trained model and train it for 4 epochs while the weights of the pre-trained model are frozen with a learning rate of 10^{-3}. For the second step, we train the fully connected layer along with the last layer of the pre-trained model at a lower learning rate of $2 * 10^{-4}$ for one more epoch.

Inspired by the approach taken by Pahwa and Pahwa (2023), who demonstrated that the performance of fine-tuned cross-encoders is comparable to that of GPT-3.5 in the few-shot setting, we experiment with the sentence-transformer model BioBERT[3] trained on 6 benchmark NLI datasets (Deka et al., 2022) for sentence similarity tasks.

[3] pritamdeka/BioBERT-mnli-snli-scinli-scitail-mednli-stsb

We train a cross-encoder model using the sentence embeddings from BioBERT on the NLI4CT train data for sentence-pair classification for 20 epochs.

We also utilized SciFive (Phan et al., 2021), a T5 model designed for biomedical literature-related tasks. This model was pre-trained on a large corpus of PubMed abstracts and PubMed Central full-text articles from biomedical and life sciences domains. It achieved state-of-the-art results on the MedNLI benchmark dataset (Romanov and Shivade, 2018). We used the SciFive model trained on MedNLI[4] in zero-shot setting to predict the entailment relations for the NLI4CT data.

4.2 Large Language Models

LLMs have achieved promising results in biomedical tasks, such as named entity recognition, relation extraction, text classification, and question answering (Jahan et al., 2023). We conducted experiments using LLaMa-2 7B[5] (Touvron et al., 2023), Mistral 7B[6] (Jiang et al., 2023) and SOLAR (Kim et al., 2023), which have shown promising results in various language tasks. LLaMa-2 is a competitive model that has performed well across multiple benchmarks such as commonsense reasoning, word knowledge, and reading comprehension. Mistral, on the other hand, has surpassed LLaMa-2 in all the tested benchmarks. We choose SOLAR-Instruct since it is a state-of-the-art model that is instruction-tuned specifically to have improved mathematical capabilities, with rephrased examples using a similar process to MetaMath (Yu et al., 2023). The team behind SOLAR developed a unique scaling technique, called depth up-scaling (DUS), which combines architectural changes with continued pre-training and obtained better results than larger models like Mixtral (Kim et al., 2023).

LLaMa-2 and Mistral were evaluated only in zero-shot settings on the NLI4CT test data. The same prompt was used in the experiments, regardless of the section the statement was referring to and it is presented in Appendix B. While the experiments from LLaMa-2 and Mistral were using the entire sections of CTRs and statements, we had a different approach for SOLAR, which involved CTRs summarization. We expand on the methodology below.

Section content summarization approach. This approach consists of two stages. First, we summarize the text of each section to reduce the number of tokens. Next, we perform the classification task on the shortened text using two-shot prompting with examples from the training set. Both stages of this pipeline use the SOLAR 10.7B Instruct v1.0 model (Kim et al., 2023) in GGUF format[7] with 5-bit quantization (Q5_K_M) using llama.cpp.

CTR summarization. We summarize the sections of each CTR in the evaluated datasets (train, development, test) to a maximum of 350 tokens. The main reason for performing summarization is to provide the model with a shorter context and a task that is easier to solve. To validate this statement, we perform preliminary runs on the development set on Single CTRs for the Results section. We obtain an F1-score of 0.55 on single results CTR without a summary, while we are able to reach 0.63-0.72 F1-score with the summarization approach.

Another motivating factor is the time required to run the inference in order to perform multiple experiments. Full CTR inference for one example can take up to 20-30 seconds. Conversely, shortened CTRs are evaluated in roughly 5 seconds. As generating summaries is a one-time cost, the time difference is compensated for after a few iterations. Evaluating on CTR summaries instead of the full CTR allowed us to dedicate more time in designing and refining prompts used for the entailment task. For these reasons, we do not conduct additional experiments on full CTRs.

The obvious drawback of summarization is the risk of discarding essential information. In the initial experiments for this approach, we tried to mitigate this by conditioning the summary to be related to the hypothesis statement. Unfortunately, this strategy caused the model to include the statement in the summary and at times even output contradictory phrases. Moreover, we did not try to continue with contextualized summaries because it would require generating a new summary for each statement instead of a summary for each CTR section. Given that the evaluation relies on using the same section with multiple statements, we need to generate only one summary per section if the summary does not depend on the hypothesis.

Inference. We solve the entailment task through zero-shot and two-shot prompting. As some statements might contain irrelevant sentences, we only keep the first sentence that contains the word "trial".

[4] razent/SciFive-large-Pubmed_PMC-MedNLI
[5] meta-llama/Llama-2-7b
[6] mistralai/Mistral-7B-Instruct-v0.2
[7] TheBloke/SOLAR-10.7B-Instruct-v1.0-GGUF

We are aware that it is not an ideal approach and we lose important information in some examples where 2 or more sentences are crucial for the task. We found that the model works better if we keep only the "trial" sentence rather than not performing this step at all. We also tried a more robust approach by asking the model to extract the relevant sentences from the hypothesis. While this tackles the issues encountered by our simple heuristic mentioned before, we limit our system setup to sentence splitting since it is significantly faster.

We limit the output grammar of the model to only "Yes" and "No" to ease processing. For each of the four sections and example types (Single or Comparison), we apply different prompts for summarization and evaluation. The final prompt templates are listed in Appendix B. The advantage of this approach is that we can analyze and tune the prompts independently for each section, without running the inference step for the whole dataset.

5 Results

Model	Setting	F1	Faithfulness	Consistency
PubMedBERT	fine-tuned	0.63	0.53	0.62
XLM-RoBERTa	fine-tuned	0.63	0.55	0.63
Deberta V3 Large	fine-tuned	0.63	0.54	0.62
BioBERT	fine-tuned	0.65	0.52	0.63
SciFive Pubmed PMC	zero-shot	0.56	0.61	0.56
Llama-2 7B	zero-shot	0.65	0.19	0.44
Mistral 7B	zero-shot	0.65	0.18	0.44
Mistral 7B Instruct-0.2	zero-shot	**0.72**	0.68	0.66
SOLAR 10B *	zero-shot	0.63	**0.90**	**0.72**
SOLAR 10B	few-shot	0.71	0.83	**0.72**

Table 1: Results of our submissions for the SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. Best results are presented in **bold**, and second-best results are presented in underline. Results with asterisk (*) were not submitted.

The official results of our team can be found in Table 1. Our results indicate that using pre-trained models on clinical text does not significantly improve the performance on this particular task, despite research confirming that domain-specific language model pre-training can be beneficial for other biomedical tasks (Gu et al., 2022). In line with last year's findings (Jullien et al., 2023b), we observe that instruction-tuned models pre-trained on generic datasets perform better than discriminative models pre-trained on biomedical datasets. With respect to LLaMa-2 and Mistral models used in zero-shot settings, they achieve high F1 scores of 0.65 and 0.72. However, these models are not robust enough and achieve low performance on Faithfulness and Consistency metrics, which are the metrics the organizers focused on. We further expand on the results of our best-performing model, SOLAR.

Control set. We obtain an F1-score of 0.71, with the highest score for comparisons of adverse events (0.79 F1) and the lowest score for comparisons of interventions (0.62 F1). Our team reaches the 16th place out of 32 participants in the official leaderboard. Compared to last year, we would be ranked on 5th place while using little to no training data and modest computational resources. Similar to last year, we report a higher Recall (0.73) than Precision (0.70).

Contrast set. Our system achieves a Faithfulness score of 0.83 (10th place out of 32 teams) and a Consistency score of 0.72 (14th place out of 32 teams). This shows that the model is more reliable when dealing with semantic-altering transformations compared to semantic-preserving changes, with only one CTR section having a faithfulness score below 0.79 (eligibility comparisons - 0.71). We observe a similar behavior in 22 other submissions where faithfulness is higher than consistency.

6 Error Analysis and Discussion

In this section, we analyze the types of errors made by the SOLAR model according to the provided metrics based on each intervention target.

Definition interventions. These interventions simply append a sentence to the statement. The model is capable of extracting the relevant sentence containing references to clinical trials if asked explicitly through a separate pre-processing step, but this incurs an additional runtime cost. Ultimately, we tackle this issue with a simple heuristic (see the inference details in 4.2).

Numerical interventions. Even though the SOLAR model has been tuned for mathematical reasoning, we identified several shortcomings. The model performs poorly with measurement units that express the same quantity in different ways. It appears to understand the meaning of symbols (e.g. "positive" instead of "+"), but if domain-specific acronyms are lowercase instead of uppercase (e.g. "hr", "her2", "mcs", "pdr"), the prediction changes.

Another interesting example is related to how entailment is affected when numbers have similar semantic meanings in other contexts. In one of the intervention trials, it is specified that a treatment

cycle takes 21 days. The statement asks whether the trial has a 30-day cycle, with the model classifying it as entailment. Slightly tweaking the number in the statement reveals that this is not regarded as entailment if the values do not match semantically: only the pairs 21-28, 21-30 and 21-31 are considered entailment; the other pairs from 21-26 to 21-34 are classified as contradiction. Nonetheless, if we change the number in the trial (21) instead of the statement, it always predicts an entailment, even when the value is nonsensical. This appears to be a drawback of long contexts as the issue only manifests with 2-shot prompting.

For numerical paraphrasing, we have 0.67 Consistency, 0.54 F1-score, 0.63 Recall and 0.47 Precision. Conversely, for numerical contradictions, the model obtains 0.80 Faithfulness, 0.85 Consistency, 0.91 F1, 1.0 Recall and 0.83 Precision[8].

Paraphrasing and contradiction interventions. We notice a similar behavior to the numerical interventions: very high scores for contradictions, but significantly lower results for rephrases. Thus, paraphrases have 0.70 Consistency, 0.69 F1, 0.72 Recall, and 0.66 Precision, while contradictions have 0.83 Faithfulness, 0.78 Consistency, 0.89 F1, 1.0 Recall, and 0.80 Precision.

Our results suggest the model is overly sensitive to any wording change between the hypothesis and premise, mistakenly interpreting them as conflicting. This also explains the perfect recall scores.

In the remainder of this section, we present a few high-level remarks related to our design decisions.

Few-shot prompting might improve results compared to zero-shot prompting. We used two examples from the training set, an entailment and a contradiction. This approach improved the results on average on the development set in terms of F1, regardless of model, model size or summarization configuration. Adding more examples to few-shot prompting might further increase the results at the expense of slower inference speed. However, with 6-shot prompting, the performance degrades even if we do not reach the context limit of 4096 tokens. We assume this is caused by the complexity of the task and the model "forgetting" what task it must solve. We do not further explore 6-shot prompting because of poor preliminary performance and increased runtime on a subset of the development data.

[8]We adapted the evaluation script to use `pos_label` as 0 for contradictions and numerical contradictions.

On the test set, it appears that few-shot prompting degrades the results in terms of Faithfulness and Contradiction, although it improves paraphrasing scores. We argue that a higher Faithfulness score in itself does not imply better results because the model could simply predict more Contradictions when the input is altered.

Instruction order is significant. We observed the best results when the instructions were placed at the start of the prompt, followed by the CTR (or CTR summary) and then the hypothesis. This strategy constantly provides improved predictions across most of the evaluated prompts. There is a drop of 4 F1 percentage points on the development set when the hypothesis is placed before the CTR summary in the prompt template. Separately, the predictions are also affected if the instruction is placed at the end of the prompt, after the hypothesis. Repeating the instruction before and after the CTR-hypothesis pair is as effective as simply placing the instruction before the CTR text.

Larger models obtain better results, but smaller models are still useful for prototyping. We use the 3-bit quantization version of the same SOLAR model (Q3_K_M) to experiment with more prompts, taking advantage of faster inference times. This approach has been very useful in designing the summarization prompts because the summarization step is the most expensive one in terms of computational resources. We also experimented with hybrid systems, where the summaries are generated with a smaller model and the inference task is done by a larger model. The performance of the hybrid strategy is comparable to running the entire pipeline with the larger model.

There are no hard constraints for summaries prompts, as long as they do not depend on the hypothesis. There appears to be no substantial difference in the final results when changing the prompt used to generate CTR summaries. We apply some of the following restrictions in each summarization prompt: use abbreviations, avoid verbs, use short sentences, be brief, maximum N words. Rephrasing the prompt does not seem to have a relevant impact for this task. As previously mentioned, it is paramount that the summaries preserve the meaning of the original text. The absence of relevant information in summaries is a major source of errors in our system. Unfortunately, for most sections the model was unable to create contextualized summaries conditioned by the hypothesis without mentioning the hypothesis in the summary.

For the Results section with a single CTR, conditioning the summary on the hypothesis looked promising. However, the next example from the test set confirms our concerns about this strategy, where a single summary for multiple hypotheses is not appropriate due to conditioning on the first hypothesis in the dataset. Our generated summary is: "There is no information in the given CTR report that relates to the statement about all patients treated with GTx-024 1mg gaining lean body mass over a 10 year period". The model is distracted by the "10 year period" from one of the hypotheses, altering the original meaning completely, even though the word "year" does not appear anywhere in the initial CTR section. See appendix A.1 for the full CTR text and associated hypotheses.

7 Conclusions and Future Work

In this paper, we presented our approach for the SemEval 2024 Task 2 (Jullien et al., 2024) aimed at understanding large language models behavior in clinical contexts. We explored several types of models and prompting techniques in order to determine whether fine-tuning is more feasible than zero-shot or few-shot prompting in a limited resource setting.

Our findings suggest that, while LLMs exhibit remarkable clinical NLI capabilities at a surface level, the proposed metrics and interventions uncover a tendency of the models to take shortcuts and rely on simple heuristics, especially when faced with semantic-preserving changes. We intend to investigate further methods of evaluating the reliability of large language models in future work.

To address the inherent weak numerical reasoning of our model (and all generative models), a promising strategy is to offload complex mathematical hypotheses to a specialized model like xVal (Golkar et al., 2023). This approach involves representing numbers as individual digits (e.g., 123 becomes ["1", "2", "3"]), replacing them with a generic [NUM] token, and scaling the token according to the original numerical value. Their results showed a 70-fold improvement over standard models. We can extend this strategy to create an ensemble where other weaknesses of our model (like rephrases) are offloaded to specialized models.

Acknowledgements

This work was partially supported by a grant on Machine Reading Comprehension from Accenture Labs and by the POCIDIF project in Action 1.2. "Romanian Hub for Artificial Intelligence".

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Surabhi Datta, Kyeryoung Lee, Hunki Paek, Frank J Manion, Nneka Ofoegbu, Jingcheng Du, Ying Li, Liang-Chin Huang, Jingqi Wang, Bin Lin, et al. 2024. AutoCriteria: a generalizable clinical trial eligibility criteria extraction system powered by large language models. *Journal of the American Medical Informatics Association*, 31(2):375–385.

Pritam Deka, Anna Jurek-Loughrey, and Deepak P. 2022. Evidence extraction to validate medical claims in fake news detection. In *International Conference on Health Information Science*, pages 3–15. Springer.

Steven Y. Feng, Vivek Khetan, Bogdan Sacaleanu, Anatole Gershman, and Eduard Hovy. 2023. CHARD: Clinical health-aware reasoning across dimensions for text generation models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 313–327, Dubrovnik, Croatia. Association for Computational Linguistics.

Lawrence M Friedman, Curt D Furberg, David L DeMets, David M Reboussin, and Christopher B Granger. 2015. *Fundamentals of clinical trials*. Springer.

Siavash Golkar, Mariel Pettee, Michael Eickenberg, Alberto Bietti, Miles Cranmer, Geraud Krawezik, Francois Lanusse, Michael McCabe, Ruben Ohana, Liam Parker, Bruno Régaldo-Saint Blancard, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. 2023. xVal: A continuous number encoding for large language models.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3:1–23.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Huang. 2023. A comprehensive evaluation of large language models on benchmark biomedical text processing tasks. *arXiv preprint arXiv:2310.04270*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2023. SOLAR 10.7B: Scaling large language models with simple yet effective depth up-scaling.

Dave Makhervaks, Plia Gillis, and Kira Radinsky. 2023. Clinical contradiction detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1248–1263, Singapore. Association for Computational Linguistics.

Bhavish Pahwa and Bhavika Pahwa. 2023. BpHigh at SemEval-2023 task 7: Can fine-tuned cross-encoders outperform GPT-3.5 in NLI tasks on clinical trial data? In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1936–1944, Toronto, Canada. Association for Computational Linguistics.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. Med-HALT: Medical domain hallucination test for large language models. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 314–334, Singapore. Association for Computational Linguistics.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. SciFive: a text-to-text transformer model for biomedical literature.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification?

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Zifeng Wang, Cao Xiao, and Jimeng Sun. 2023. AutoTrial: Prompting language models for clinical trial design. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12461–12472, Singapore. Association for Computational Linguistics.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. MetaMath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

A Additional examples

A.1 Summary mismatch example

The Results section of the CTR with ID "NCT00467844" is the following:

> Outcome Measurement: The Efficacy of GTx-024 on Total Body Lean Mass. Change in total body lean mass as measured by dual energy x-ray absorptiometry (DEXA)from baseline to 4 months. Time frame: Baseline to Four Months.
>
> Results 1: Arm/Group Title: GTx-024 1 mg Arm/Group Description: [Not Specified] Overall Number of Participants Analyzed: 32 Median (Full Range) Unit of Measure: kg 1.55 (-2.06 to 12.64)
>
> Results 2: Arm/Group Title: GTx-024 3 mg Arm/Group Description: [Not Specified]

Overall Number of Participants Analyzed: 34 Median (Full Range) Unit of Measure: kg 0.98 (-4.84 to 11.54)

The summary is conditioned on the following hypothesis: "all patients treated with gtx-024 1mg in the primary trial gained lean body mass over a 10 year period". However, other hypotheses are concerned with other quantities: "at least one patient treated with GTx-024 1mg in the primary trial gained over 10 kilos of Lean body Mass". For the latter, our summary is misleading because the "10 kilos" information is missing. This could be mitigated by refraining to summarize short sections.

B Prompt templates

The prompt templates used to obtain the final leaderboard results for SOLAR are shown in tables 4 and 5.

We started with a single prompt template for all sections and summaries. When the results did not further improve, we analyzed the F1-score of each section, shown in Tables 2 and 3. Due to time constraints, we only create summaries for the first 50 examples in the train set.

While the single CTR summaries for the results section depend on the hypothesis, due to an implementation choice, all examples use the same CTR summary, regardless of the hypothesis (only the first hypothesis is used). We believe that this issue is not essential, since all the hypotheses for a CTR focus on the same information.

Initial summary prompt: "Instruction: You are given a clinical trial report. You must summarize the report. Use abbreviations. Be brief. Report: {premise}. Summary (max 350 words):".

Initial evaluation prompt: "Instruction: You are given a Clinical Trial Report and a hypothesis. ##Report: {premise}. ##Hypothesis: {hypothesis}. ##Can the hypothesis be inferred from the report? Respond only with Yes or No. ##Response (Yes or No):". This prompt was also used for our experiments with LLaMa-2 and Mistral.

C Infrastructure

In terms of infrastructure, we use a system with 16 GB RAM and an NVIDIA GTX 1060 MQ GPU with 6 GB VRAM. Out of the 5500 examples on the test set, this method only requires generating summaries for 251 examples comprising different CTR sections. On the development set, we need to create 100 summaries for the 200 samples, making the summarization step more expensive in this regard.

Section	F1
Eligibility (Single)	0.6637
Eligibility (Comparison)	0.5274
Intervention (Single)	0.7306
Intervention (Comparison)	0.7751
Results (Single)	0.7141
Results (Comparison)	0.7525
Adverse events (Single)	0.7141
Adverse events (Comparison)	0.6805

Table 2: Initial results for the train set (first 50 examples)

Section	F1
Eligibility (Single)	0.8178
Eligibility (Comparison)	0.8285
Intervention (Single)	0.7678
Intervention (Comparison)	0.6000
Results (Single)	0.7368
Results (Comparison)	0.7749
Adverse events (Single)	0.5835
Adverse events (Comparison)	0.6969

Table 3: Initial results for the development set

The inference time for a summary varies with the length of the CTR section, with a minimum time of 30 seconds per sample and a total time of about 7 hours, but it should be noted that this stage is a one-time cost. The inference time for one example is approximately 5 seconds, meaning that the evaluation on the test set takes roughly 9 hours, with the total time reaching 12 hours when applying 2-shot prompting, since the sequence length increases.

Section (CTR type)	Prompt
Eligibility (Single)	Instruction: You are given clinical trial criteria and a statement that may or may not be contradictory. Regarding the inclusion and exclusion criteria, is the statement correct? Respond only with Yes or No. ## Criteria: {premise}. ## Statement: {hypothesis}. ## Response (Yes or No):
Eligibility (Comparison)	Instruction: You are given clinical trial criteria for a primary and a secondary trial, and a statement. Regarding the inclusion and exclusion criteria, is the statement correct for each trial? Respond only with Yes or No. ## Criteria: {premise}. ## Statement: {hypothesis}. ## Response (Yes or No):
Intervention (Single)	Instruction: You are given a CTR and a statement. Can the statement be deduced from the CTR? Focus on the interventions. Respond only with Yes or No. ##CTR: {premise}. ##Statement: {hypothesis}. ##Response (Yes or No):
Intervention (Comparison)	(same prompt template as single CTR for interventions)
Results (Single)	Instruction: You are given the results of a CTR and a statement. Can the statement be deduced from the CTR in terms of number of participants, measures and results? Respond only with Yes or No. ##CTR: {premise}. ##Statement: {hypothesis}. ##Response (Yes or No):
Results (Comparison)	Instruction: You are given the results of a CTR and a statement. Can the statement be deduced from the CTR? Respond only with Yes or No. ##CTR: {premise}. ##Statement: {hypothesis}. ##Response (Yes or No):
Adverse events (Single)	Instruction: You are given a CTR and a statement. Regarding the adverse events, signs and symptoms observed in the CTR, is the statement correct? Respond only with Yes or No. ##CTR: {premise}. ##Statement: {hypothesis}. ##Response (Yes or No):
Adverse events (Comparison)	(same prompt template as single CTR for adverse events)

Table 4: Evaluation templates for each CTR section

Section (CTR type)	Prompt
Eligibility (Single)	Instruction: You are given the eligibility criteria for a clinical trial report. You must summarize the report focusing on inclusion and exclusion criteria. Report: {premise}. Use short sentences. Summary:
Eligibility (Comparison)	(same prompt template as single CTR for eligibility)
Intervention (Single)	Instruction: You are given the intervention information for a clinical trial report. Each report contains 1-2 cohorts, which receive different treatments, or have different characteristics. You must summarize the report focusing on the type, dosage, frequency, and duration of treatments being studied. Report: {premise}. Use short sentences to group by cohort. Summary:
Intervention (Comparison)	Instruction: You are given the intervention information for a clinical trial report. Each report contains 1-2 cohorts, which receive different treatments, or have different characteristics. You must summarize the report focusing on the type, dosage, frequency, and duration of treatments being studied. Report: {premise}. Use short sentences to group by cohort and group by primary trial and secondary trial. Summary:
Results (Single)	Instruction: You are given the results of a CTR and a statement. Extract all the relevant information from the CTR that is related to the statement. Report: {premise}. Statement: {hypothesis}. Answer:
Results (Comparison)	Instruction: You are given the results of two clinical trials. Each trial contains 1-2 cohorts, which receive different treatments, or have different characteristics. You must summarize the report for each trial, focusing on number of participants, outcome measures, units, results. Report: {premise}. Use short sentences and keep all numeric values. Summary:
Adverse events (Single)	Instruction: You are given the adverse events of a clinical trial report. Each report contains 1-2 cohorts, which receive different treatments, or have different characteristics. You must summarize the report focusing on the adverse events, signs and symptoms observed in patients. Report: {premise}. Use short sentences. Summary:
Adverse events (Comparison)	Instruction: You are given the adverse events of a clinical trial report. Each report contains 1-2 cohorts, which receive different treatments, or have different characteristics. You must summarize the report focusing on the adverse events, signs and symptoms observed in patients. Report: {premise}. Use short sentences and group by primary trial and secondary trial. Summary:

Table 5: Summarization templates for each CTR section. The results section (single CTR) is the only one for which summaries depend on the hypothesis due to lack of time to rerun the summaries for the test set.

Fralak at SemEval-2024 Task 4: combining RNN-generated hierarchy paths with simple neural nets for hierarchical multilabel text classification in a multilingual zero-shot setting

Katarina Laken
Fondazione Bruno Kessler / Trento, Italy
Universidade de Santiago de Compostela / Santiago de Compostela, Spain
alaken@fbk.eu

Abstract

This paper describes the submission of team fralak for subtask 1 of task 4 of the Semeval-2024 shared task: 'Multilingual detection of persuasion techniques in memes'. The first subtask included only the textual content of the memes. We restructured the labels into strings that showed the full path through the hierarchy. The system includes an RNN module that is trained to generate these strings. This module was then incorporated in an ensemble model with 2 more models consisting of basic fully connected networks. Although our model did not perform particularly well on the English only setting, we found that it generalized better to other languages in a zero-shot context than most other models. Some additional experiments were performed to explain this. Findings suggest that the RNN generating the restructured labels generalized well across languages, but preprocessing did not seem to play a role. We conclude by giving suggestions for future improvements of our core idea.

1 Introduction

Task 4 of the Semeval 2024 workshop deals with the identification of persuasion techniques in meme data (Dimitrov et al., 2024). Subtask 1 regarded only the textual content of the memes. Training, validation and development data is only available in English, but the test phase includes data in three more languages (Bulgarian, North Macedonian and Arabic) for multilingual zero-shot classification. The 20 persuasion techniques are organized in an hierarchical directed acyclic graph (available on the task website). Each meme can have zero, one or multiple persuasion techniques associated with it, making this a hierarchical multilabel classification problem. Assigning a parent node of the target label results in partial points.

Our system (team fralak) implements an ensemble model including a seq2seq module, using some innovations to avoid common pitfalls and exploit the hierarchy information. Our approach transforms the problem by restructuring the labels into strings in a way that captures all possible paths through the hierarchy and uses these as target sequences to train a RNN that learns the relationships between the labels on different levels. It combines the power of a RNN with a simple fully connected architecture. These non-sequential modules are also expected to mitigate the error propagation effect (also called exposure bias), where a wrongly predicted label in the beginning of the generated sequence results in more errors down the line (Xiao et al., 2021). RNNs for multilabel classification also depend on the ordering of the labels, even though the class labels are essentially an unordered set (Wang et al., 2021a). We address this by sorting the labels by frequency. The textual content of the memes is represented using multilingual sentence embeddings. Although we only participated in subtask 1, our architecture can easily be expanded to also take into account the visual content of the meme (subtask 2a+b)[1]. Our system performed below average on the English-only test set, but generalized better than most other systems. The goal of this paper is to explain our methodology and system architecture (sections 2 and 3) and explore why the system performs relatively well at the zero-shot task (section 5).

1.1 Background

Hierarchical multilabel classification is applied in many domains, from biology (genomics) (Romero et al., 2023; Wang et al., 2021b) to the classification of images (Lanchantin et al., 2021) or text data (Xiao et al., 2021; Omar et al., 2021). A challenge of this type of data is that the data is virtually always unbalanced on all levels of the hierarchy (Tarekegn et al., 2021). Labels also tend to be correlated.

[1] although the performance of the system on these multimodal tasks remains to be seen

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 596–601

There are several approaches to hierarchical multilabel classification. Some studies transform the problem, for example by creating a chain of binary classifiers, whereas others adapt the classification algorithm (Bogatinovski et al., 2022). Some approaches construct a model for each label, but this becomes very computationally expensive: once the amount of labels grows, it is difficult for labels with very few instances, and it is difficult to capture relationships between the labels (Chen and Ren, 2021). The hierarchy can be leveraged for classification. For example, Giunchiglia and Lukasiewicz (2020) use prediction on the lower classes in the hierarchy to make predictions on the upper ones. Seq2seq models are popular for multilabel classification (Chen and Ren, 2021; Chen et al., 2023; Huang et al., 2021). The main idea behind the employment of seq2seq models is that they are able to capture the correlations between labels (Chen and Ren, 2021). Huang et al. (2021) found that a seq2seq model using a biLSTM outperformed other SOTA approaches using chains of classifiers.

The past years have seen the rise of transfer learning, where some model is used for the classification of a different type of data than the data it was trained on (Iman et al., 2023). A common approach to multilingual transfer learning is the use of mapping words or sentences to vectors in a vector space that aligns embeddings for different languages. Training some model on these representations in language A then allows it to make predictions about data in unseen data B, as long as its embeddings are meaningfully mapped to the same vector space (Reimers and Gurevych, 2019).

2 Methodology

We aimed to implement rather simple NN modules in order to explore their usefulness for a complicated task like this. The main idea behind our system is to transform the labels into strings that reflect the hierarchical acyclic graph containing the different persuasion techniques. These are be used to train an RNN that is supposed to learn the relationships between both the labels and the different levels of the hierarchy. We expect that the relations between labels are a feature that generalizes especially well across languages, making our approach especially adapt for multilingual zero-shot learning for this specific task.

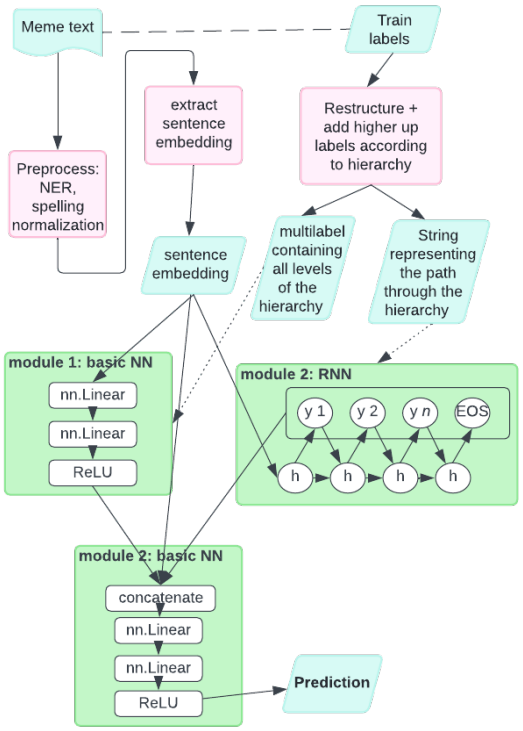


Figure 1: The hierarchical directed acyclic graph containing the persuasion techniques

2.1 Preprocessing

The preprocessing consisted of three main steps: spelling normalization, named entity recognition (NER), and adding the sentence embeddings. Since we wanted the system to be able to be applied to other languages as well, preprocessing was limited to some regular expressions capturing characters that repeated more than twice, irregular white spaces, and regularizing different kinds of *haha*'s to a simple 'haha'[2].

The NER was performed using a pretrained multilingual model for token classification by Babel[3], accessed through the Huggingface API. We compiled a list of the 10 people most commonly occurring in the training data and made a dictionary that 'translated' all of their names to one token (so 'Barack Obama' or 'Barack Hussein Obama' etc. would both be 'translated' to 'Obama'). All person entities (recognized with a certainty of over .8) that did not appear in this list were replaced by the name 'Mark', in order for them to be mapped to

[2] t = re.sub(r'[AaHhJjXxAa]*[HhJjXx]?[AaAa]+[HhJjXx] + [AaAa]+[AaHhJjXxAa]*', 'haha', t) ; the double a's are because one is the Cyrillic A and the other is the Latin A

[3] https://huggingface.co/Babelscape/wikineural-multilingual-ner

some kind of baseline name rather than the OOV token[4].

Meme data is expected to contain a lot of non-normative language. We chose to represent it using multilingual sentence embeddings, as these are typically better at dealing with OOV tokens. We used the multilingual variation of the sentence embeddings by (Reimers and Gurevych, 2019)[5].

These embeddings were used as the input for a neural architecture consisting of three separate modules (see 1). Module 1 consisted of a simple neural network (of hidden size 128), trained over 45 epochs, with one input layer, one output layer, and a ReLU activation. The input consisted of the sentence embedding for the meme; the output was a simple multiclass classification with one output for each activated node (meaning the target label and all of its progenitore nodes).

2.2 Restructuring of the labels

The central innovation of our approach is the inclusion of hierarchy information by way of transforming the labels to strings reflecting all possible paths through the hierarchy. Module 2 was an RNN that learnt to generate a sequence reflecting the labels and the hierarchy they were embedded in. First, all labels were sorted by frequency in the training data; multiword labels were turned into one-word labels (for example, 'thought terminating cliché' became 'cliché'). We then added the labels from the levels above them (as represented in the label hierarchy graph). As the hierarchy has different levels, this means that every meme was doubled or tripled in the training data, but with different labels. For instance:

Sequence: VISIT RUSSIA\n\nBEFORE RUSSIA VISITS YOU
Label 1: *Only labels*
'repetition and black and appeal EOS'
Label 2: *Labels + red level*
'logos namely repetition and logos namely black and logos and pathos namely appeal EOS'
Label 3: *Labels + red level + blue level*
'logos namely repetition and logos namely reasoning namely black and logos namely justification and pathos namely appeal EOS'
Label 4: *Labels + red level + blue level + green level*
'logos namely repetition and logos namely reasoning namely simplification namely black and logos namely justification and pathos namely appeal EOS'

The idea of this doubling of labels was that higher-up levels would appear more often and thus become more likely to be predicted by the module. However, preliminary testing showed that it made hardly any difference to use only labels of type 4 or all kinds of labels, likely because lower level labels inherently appear less often due to them governing less nodes.

2.3 System architecture

A simple RNN (hidden size = 128) was trained over 25 epochs to generate restructured labels. The model generates labels either until the max string length (manually set to 50) was reached, or until the EOS token was generated[6]. This module was supposed to learn the relationships between labels both at the same and at different levels of the hierarchy; we expected this knowledge to transfer rather well to the unseen multilingual data.

The final module (module 3 in figure 1) concatenates the meme embedding with the outputs of the modules 1 and 2 and the meme embedding and passes it through two fully connected layers (hidden size = 128) and a ReLU activation function (dropout = 0.2). This module, that was trained over 50 epochs, outputs the final prediction of the labels.

3 Experimental setup

The training+validation data consisted of 7,500 memes. After restructuring the labels this gave us 21,968 training instances. Including the development data (1,000 memes) resulted in 24,664 instances spread over 8,500 memes. All of these instances were in English. As typical for multilabel settings, the class labels are extremely unbalanced: the most common label, *Smears*, occurs 1,990 times in the training data, whereas the least common label, *Intentional vagueness*, occurred only 21 times. Our teams original test submission was only trained on the train and validation data, as we used the development data to validate our approach, but we added the development data in subsequent experiments as we theorized that having

[4]this step did not take into account different alphabets
[5]Accessed through the Huggingface API (model card)
[6]in the training of the module we used teacher enforcement, so there was no maximum string length; however, we did use this when generating the training data for the final module, so the final module had not seen RNN-generated inputs of over 50.

Type	Rank	F1	P	R
Dev (English)	27/33	0.55	0.47	0.66
Test English	25/33	0.56	0.48	0.67
Test Bulgarian	10/20	0.46	0.37	0.61
Test North Maced.	4/20	0.46	0.36	0.66
Test Arabic	3/20	0.43	0.31	0.70

Table 1: Table showing the main results of our official submissions. The rank x/y shows our position x and the total amount of teams that made a test submission y

more training data would give more robust results; as we did not do additional finetuning for the post-hoc experiments, no development set was used (see section 5). Due to the way the Semeval challenge was set up, the validation set was a dataset that was available from the beginning, whereas the gold labels for the development set only became available a couple of weeks before the test submission closed; we only used the validation set for some preliminary testing and setup, after which it was joined with the test set. All results are reported on the test data.

The modules were trained separately, but on the same data. We conducted some preliminary experiments training module 1 and 2 on 75% of the data and module 2 on the remaining 25% (random split) but this led to a drop in performance. The optimal amount of epochs for each module was decided based on plots of the average loss per epoch. Each separate module took less than 30 minutes to train on an Apple M3 8-core CPU. We used an Adam optimizer with a learning rate of 1e-3. Modules 1 and 3 were trained with a CrossEntropyLoss; module 2 with a SmoothL1Loss.

The task evaluation metric was the hierarchical-F1, calculated using hierarchical precision and recall (Kiritchenko et al., 2006). This measures gives partial points for assigning a label higher up in the hierarchy, and full points for assigning the specific technique.

4 Results

Table 1 shows the outcomes of our official submissions on the test and dev sets (before 1/2/2024). There was originally an issue with the Arabic gold labels; the reported scores correspond to the corrected version of the gold labels. For the English data, the test results were very much in line with the results on the dev leader board, with only 0.01 point difference in the hierarchical-F1. The results on the zero-shot test submissions were more surprising: although the F1 was (expectedly) lower

Model description	Language	F1	P	R
As test submission[7]	Eng.	0.56	0.45	**0.72**
	Bulg.	0.46	0.35	**0.68**
	Maced.	0.45	0.33	**0.70**
	Arabic	0.40	0.28	**0.71**
No NER	Eng.	0.55	0.46	0.68
	Bulg.	0.47	0.37	0.64
	Maced.	0.46	0.36	0.64
	Arabic	0.42	0.30	0.68
No preprocessing[8]	Eng.	**0.57**	0.49	0.67
	Bulg.	0.47	0.38	0.62
	Maced.	0.46	0.36	0.63
	Arabic	0.42	0.31	0.64
Only module 1	Eng.	0.53	0.48	0.6
	Bulg.	0.44	0.36	0.56
	Maced.	0.42	0.35	0.55
	Arabic	0.40	0.32	0.56
Only module 2	Eng.	0.46	**0.54**	0.39
	Bulg.	0.37	**0.41**	0.34
	Maced.	0.32	**0.39**	0.27
	Arabic	0.38	**0.34**	0.42
MLL[9] RNN = 100	Eng.	0.54	0.50	0.58
	Bulg.	**0.48**	0.40	0.61
	Maced.	**0.48**	0.38	0.65
	Arabic	**0.42**	0.30	0.68

Table 2: Table showing the results of subsequent experiments on the test set

than for the English data, the ranking showed that the system still generalized considerably better than most other approaches. Section 5 discusses some possible explanations and describes additional experiments aimed at shining light at this question.

5 Discussion

We hypothesize that two mechanism that might have contributed to the system's generalization capacity. First, the preprocessing (particularly the NER step) might have made the model more generalizable. Second, the seq2seq RNN module to learn the labels might have been particularly good at capturing the relationships between the labels

In order to investigate these hypotheses, we ran additional experiments in which we left out parts of the system to investigate what happened to the performance. The results are summarized in table 2. All of these models were trained on train, validation and development set and tested on the test set (there was no development set as no additional fine-tuning was performed). Note that this is different from the original submission, that was trained on the training and validation set, validated on the development set, and tested on the test set; the manipulations made in the post-hoc experiments should thus be compared to the results in the upper row of table 2, which is a re-run of the model as described in section 2, but trained on the 1000 more memes of

the development set.

Our first hypothesis was that the preprocessing, especially the NER, helped the system generalize better to unseen languages. However, this seems not to be the case. Taking out only the NER module let to a slight drop in performance in English, but a better performance in the other languages. A possible explanation is the non-ubiquity of the name Mark: replacing people with 'Mark' might not actually be helpful if 'Mark' is not adapted to the specific language. Skipping all preprocessing steps (other than adding the embeddings) actually improved performance for English (even though taking out only the NER led to a drop, suggesting this might actually have been a very helpful step for the English data), but made hardly any difference for the other languages when compared to the setting without NER.

Our second hypothesis was that the RNN module was especially helpful for generalization. Either module alone performed worse than the three modules combined for all languages (apart from the first module, that reached the same F1 for Arabic), so the influence of the label-generating RNN should not be overestimated. On the other hand, when comparing the performance of module 1 with the performance of module 2, we see that the difference is the same for English and Bulgarian, and bigger for both Macedonian and Arabic. This might mean towards the second module actually being a bit more important in the zero-shot setting, but more research is required.

Our full model had remarkably high recall, but low precision. Looking at the performance of modules 1 and 2 separately suggests that this is mainly due to module 1 (and, possibly, module 3, that is very similar to module 1 in architecture). This pattern is the same across languages and modifications. This is not very surprising; erroneously generating the EOS token once makes the module stop predicting labels, and given that every training instance has an EOS token, it is very common and the chance of it being produced erroneously is relatively high. Moreover, the RNN stops generating strings when the maximum string length of 50 is reached. We thus re-ran the base model (including development model) with a maximum string length of 100 for the RNN (table 2). This resulted in the best model thus far for the zero-shot setting due to improved precision, but the performance for the English test data fell marginally. This is a further indication that the RNN module is indeed crucial to the zero-shot classification.

Adding the development data (i.e. training the model on more data) seems marginally helpful for English (+0.01 point F1) but marginally unhelpful for Macedonian (-0.01 point F1) and Arabic (-0.03 point F1). If the strength of our system indeed lies in it learning the relationships between the labels of the hierarchy, it is likely that a smaller amount of data was just enough to learn this, and adding more data just makes the model overfit.

6 Conclusion

This paper described the system used to generate the test submissions for subtask 1 of task 4 of SemEval 2024 'multilingual detection of persuasion techniques in memes'. We proposed a system consisting of different neural modules, the most innovative of which was an RNN that was trained to generate sequences that reflect the position of the relevant labels in the hierarchy. Our model did not perform particularly well on the English data, but compared to the other teams, it generalized unexpectedly well to other languages in a zero-shot setting. We conducted some additional experiments to find out what might have contributed to this. We found that our preprocessing steps (normalization and NER) did not make the model more generalizable, but we did find some evidence that the RNN module might have played a role as hypothesized.

We see plenty of possibilities to improve on our original idea in the future. First of all, we would like to explore the performance of different types of embeddings. We found that our system as a whole had a high recall, but a low precision; however, the RNN module showed the exact opposite pattern, having high precision and low recall. Allowing the RNN output in module 3 to be longer (up to 100 tokens) partially alleviated this problem and improved performance. We hypothesize this is because the EOS token is generated too easily. Somehow raising a barrier for the module to generate the EOS token might help to improve its recall. Implementing an attention mechanism in the final module could also help aleviate this problem. Other options to explore are a NER preprocessing step that better generalizes to other languages than just replacing people with "Marks". Finally, it would be interesting to explore the capabilities of a hierarchical-path generating RNN with more sophisticated layers (GRU or LSTM), or combined with a convolutional model.

Funding and acknowledgements

This research was carried out in the context of the HYBRIDS project. This project has received funding from the European Union´s Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101073351. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

This research was carried out at the Fondazione Bruno Kessler (Trento, Italy) and the CiTIUS (Santiago de Compostela, Spain). A special thanks to dr. Sara Tonelli and dr. Marcos García González for supervising this project, Erik Bran Marino for his comments on the paper draft, and Rafael Frade for the valuable feedback and support. Finally I would like to thank the anonymous reviewers for their time, effort and important and helpful insights.

References

Jasmin Bogatinovski, Ljupčo Todorovski, Sašo Džeroski, and Dragi Kocev. 2022. Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications*, 203:1–18.

Xiaolong Chen, Jieren Cheng, Zhixin Rong, Wenghang Xu, Shuai Hua, and Zhu Tang. 2023. Multi-label text classification based on improved seq2seq. In *International Conference on Computer Engineering and Networks*, pages 439–446. Springer.

Ziheng Chen and Jiangtao Ren. 2021. Multi-label text classification with latent word-wise label information. *Applied Intelligence*, 51:966–979.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Eleonora Giunchiglia and Thomas Lukasiewicz. 2020. Coherent hierarchical multi-label classification networks. *Advances in neural information processing systems*, 33:9662–9673.

Chenyang Huang, Amine Trabelsi, Xuebin Qin, Nawshad Farruque, Lili Mou, and Osmar R Zaiane. 2021. Seq2emo: A sequence to multi-label emotion classification model. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 4717–4724.

Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. 2023. A review of deep transfer learning and recent advancements. *Technologies*, 11(2):40.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*, pages 395–406. Springer.

Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. 2021. General multi-label image classification with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16478–16488.

Ahmed Omar, Tarek M Mahmoud, Tarek Abd-El-Hafeez, and Ahmed Mahfouz. 2021. Multi-label arabic text classification in online social networks. *Information Systems*, 100:1–18.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Miguel Romero, Felipe Kenji Nakano, Jorge Finke, Camilo Rocha, and Celine Vens. 2023. Leveraging class hierarchy for detecting missing annotations on hierarchical multi-label classification. *Computers in Biology and Medicine*, 152:106423.

Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. 2021. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965.

Ran Wang, Robert Ridley, Weiguang Qu, Xinyu Dai, et al. 2021a. A novel reasoning mechanism for multi-label text classification. *Information Processing & Management*, 58(2):102441.

Wei Wang, QiuYing Dai, Fang Li, Yi Xiong, and Dong-Qing Wei. 2021b. Mlcdforest: multi-label classification with deep forest in disease prediction for long non-coding rnas. *Briefings in Bioinformatics*, 22(3):1–11.

Yaoqiang Xiao, Yi Li, Jin Yuan, Songrui Guo, Yi Xiao, and Zhiyong Li. 2021. History-based attention in seq2seq model for multi-label text classification. *Knowledge-Based Systems*, 224:107094.

OtterlyObsessedWithSemantics at SemEval-2024 Task 4: Developing a Hierarchical Multi-Label Classification Head for Large Language Models

Julia Wunderle[†] and **Julian Schubert**[‡] and **Antonella Cacciatore**[‡]

Albin Zehe[†] and **Jan Pfister**[†] and **Andreas Hotho**[†]
Center for Artificial Intelligence and Data Science (CAIDAS)
Data Science Chair, Julius-Maximilians-Universität Würzburg (JMU)
[†]{lastname}@informatik.uni-wuerzburg.de
[‡]{firstname.lastname}@informatik.uni-wuerzburg.de

Abstract

This paper presents our approach to classifying hierarchically structured persuasion techniques used in memes for Task 4 Subtask 1 of SemEval 2024. We developed a custom classification head designed to be applied atop of a Large Language Model, reconstructing hierarchical relationships through multiple fully connected layers. This approach incorporates the decisions of foundational layers in subsequent, more fine-grained layers. To improve performance, we conducted a small hyperparameter search across various models and explored strategies for addressing uneven label distributions including weighted loss and thresholding methods. Furthermore, we extended our preprocessing to compete in the multilingual setup of the task by translating all documents into English. Finally, our system achieved third place on the English dataset and first place on the Bulgarian, North Macedonian and Arabic test datasets.

1 Introduction

Memes are widely used for communicating in the digital age, often laced with sarcasm and humor. However, beyond their role in everyday conversation, memes are increasingly recognized for their persuasive and manipulative potential. They hold power to subtly influence opinions, incite reactions, or shape public discourse and perception. Given this dual nature of memes as both funny communication tool and vehicle for manipulation, there arises a need to dissect and understand the persuasion techniques embedded within them. A proper understanding of this domain enhances the ability to reflect on and emotionally defend against manipulation. In this context, Large Language Models (LLMs) emerge as valuable assets in analyzing and deciphering the persuasive elements within memes. Their automated, rapid processing capabilities make them well-suited for parsing through vast amounts of meme data, extracting patterns, and discerning underlying features. Recognizing the importance of this topic, the SemEval 2024 Task 4 Subtask 1 focuses on identifying persuasion techniques used in memes (Dimitrov et al., 2024). The aim of the first subtask is to classify the textual content from memes into various hierarchically structured persuasion techniques. In this paper, we provide a detailed description of our system including the custom classification head we designed in order to incorporate the hierarchy of the labels. Our system was able to achieve the third place on the English test dataset. Furthermore, we outperformed all other systems on the Bulgarian, North Macedonian and Arabic test sets. In summary, (i) we created a custom classification head well-suited for hierarchical settings, (ii) developed a strategy for languages where less training data is available, (iii) analyzed the influence of different hyperparameters and strategies in the context of multi-label classification problems. Our code is publicly available[1].

2 Related Work

In the context of multi-label classification, the primary aim is to identify all relevant classes associated with a given sample. Additionally, in a hierarchical classification setting the labels are partially ordered, ranging from broader generic categories to narrowed specific instances (Kiritchenko et al., 2006). There is a large variety of approaches for this task. While earlier methods were based on tree-structures and graphs, more recent approaches rely on deep learning models (Liu et al., 2023). This section introduces various models adaptable to the task of hierarchical multi-label classification.

These authors contributed equally to this work.

[1]https://github.com/LSX-UniWue/Semeval-2024-Task-4

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 602–612

2.1 Models

Transformer models consist of an encoder and decoder which can individually be adapted for sequence classification tasks (Vaswani et al., 2017).

Encoder-only Models are well-suited for sequence classification. These models directly generate a representation of the input sequence, which is then passed through a classification head for prediction. As huggingface (Wolf et al., 2020) allows us to easily test different models, we compared a variety of encoder-only models. This includes different *BERT* (Devlin et al., 2019) and *RoBERTa* (Liu et al., 2019) models. As the memes in our dataset often contain hateful or toxic content, we include specifically pre-trained *BERT-base* models. While *hateBERT* (Caselli et al., 2021) is re-trained on explicitly hateful content from banned reddit communities, *bert-hateful-memes-expanded* (limjiayi, 2021) was fine-tuned on multiple datasets containing hateful memes.

Decoder-only Models like *LLaMA 2* can be adapted for sequence classification tasks by utilizing the logits of the last token from the input sequence (Huggingface). We evaluated the performance of the 7b and 13b parameter versions of *LLaMA 2* (Touvron et al., 2023).

3 Dataset

The organizers provided English datasets for training, validation and testing (7000/500/1500, train/val/test). Additionally a dev set (1000) was published to enable comparison of participating systems on a separate leaderboard ahead of the final submission on the test data. Each sample within these datasets consists of an unique id, the URL linking to the source of the meme, the transcribed text content and a list of associated labels. For example, the text: *Stay on high moral ground and we will win - Raphael Warnoc*, has the associated labels: *Appeal to authority* and *Glittering generalities (Virtue)*. The labels of the memes are structured hierarchically, with *Ethos*, *Pathos* and *Logos* in the first layer. In total, there are 28 labels with 20 persuasion techniques in the last layer, which we will refer to as *leaves*. It is important to note that the leaves are not distributed equally within the datasets. While *Smears* (1990), *Loaded Language* (1750) and *Name calling/Labeling* (1518) appear most frequently in the training data, *Presenting Irrelevant Data (Red Herring)* (59) and *Obfuscation, Intentional vagueness, Confusion* (21) occur most rarely. The final submissions were made on the English test dataset. In addition, the hosts released testing data for North Macedonian (259), Bulgarian (436) and Arabic (100) (Dimitrov et al., 2024).

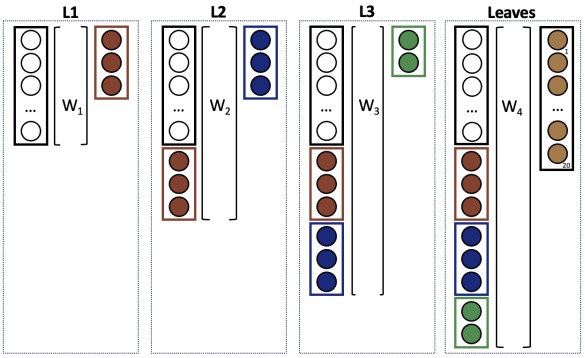


Figure 1: Illustration of our custom classification head. The depicted parts represent different layers, where **L1** corresponds to the first hierarchy layer: Ethos, Pathos, Logos. **L2** maps to the second layer, and **L3** to the third. Finally, all features are mapped to the **Leaves**. This design allows us to incorporate previously made decisions into subsequent layers. For simplicity, W_{1-4} represent fully connected layers.

4 System Overview

This section provides an in-depth description of our system. To integrate the hierarchical structure of the labels, we introduce a custom classification head that is designed to be applied atop various pre-trained Large Language Models.

4.1 Pre-Processing

We tested our system with two different pre-processing approaches: In memes, lines of text are often broken due to space limitations on the image. Therefore, we assume that most newline characters do not carry any semantic information and thus remove them in the first pre-processing variation (*cleaned*). As preliminary experiments indicated that certain LLMs might exhibit enhanced performance with all-lowercase input, the second version incorporates an additional step to convert the text to lowercase letters (*all_lower*).

4.2 Custom Classification Head

The fundamental concept of our classification head entails intuitively reconstructing the hierarchy across multiple fully connected layers. As depicted in Figure 1, the basic architecture unfolds as follows: In the initial layer (**L1**), the sequence embedding provided by the backbone LLM serves as

input, producing logits for the three highest nodes of the hierarchy, namely: Ethos, Pathos and Logos. Logits for the next layer (**L2**), Ad hominem, Justification and Reasoning, are obtained by passing a concatenation of the sequence embedding and the logits of the preceding layer through another feed-forward layer. This process is repeated for the last parent nodes, Distraction and Simplification (**L3**). Finally, the logits for the 20 individual leaf nodes (**Leaves**) are obtained using another linear layer, which incorporates the concatenation of sequence embeddings and the logits of all previous layers. Accessing decisions from upper levels of the hierarchy enables logits in the fine-grained layers to be shaped by the choices made for more foundational categories. Crucially, the loss is calculated over all nodes, not solely leaf nodes, enabling the model to learn the hierarchy effectively. The head outputs logits for all 28 labels, which are then transformed into probabilities using a sigmoid function. Lastly, the probabilities are converted into labels using thresholds, where labels with a probability above the threshold are included in the final prediction. Notably, all classes in the hierarchy, including leaf nodes with low parent probabilities and vice versa, can be predicted. This design principle ensures that decisions made at higher levels serve as guidance without imposing restrictions, thereby maintaining the autonomy of lower-level decisions within the hierarchical structure.

4.3 Loss function

We aimed to address the unequal label distribution by testing both the standard binary cross-entropy loss as well as its weighted variant. Each class was weighted depending on their inverse frequency, assigning a higher penalty to misclassifications of minority classes, with the goal of enhancing the performance of these less represented classes.

4.4 Ensemble

To further enhance the robustness of our predictions we employed an ensemble approach where we utilize majority voting across four different models. Each of these models was trained with the same hyperparameters but with different random seeds. As described above (Section 4.2) our model outputs labels for each sample. To combine the suggestions of multiple models, we experimented with various boundary levels to determine the number of model predictions needed for a label to be included in the final prediction. Our experiments revealed that requiring at least two of the four models to vote for a label is the most effective.

4.5 Handling Different Input Languages

To extend the applicability of our system for the multilingual setting, we integrated an additional pre-processing step. The provided non-English test datasets were translated into English using GPT-4 (OpenAI et al., 2023), using the following prompt: *You are a bilingual humorist, adept at translating meme text between languages while preserving the original humor, cultural nuances, and any slang or idiomatic expressions. Ensure the translation is accurate, contextually appropriate, and retains the meme's playful tone. Avoid adding explanations or additional commentary and provide only the translation.*

5 Experimental Setup

In order to approximate optimal parameters for the LLaMA 2 models, we conducted a grid-search for various BERT and RoBERTa models as these require less computational resources. During training, we utilized gradient accumulation to reach a gradient update every 128 samples. All models were trained for ten epochs, with a learning rate of either 5×10^{-4} or 5×10^{-5} and the Adam optimizer (Kingma and Ba, 2014). We further included the two different pre-processing styles as well as the binary-cross entropy loss and its weighted variation as hyperparameters. Lastly, we performed all experiments with and without our custom classification head. Training was conducted on either NVIDIA GeForce RTX 4090 or NVIDIA A100 GPUs. Due to the large size of the LLaMA 2 models, we used Low-Rank-Adaptation to greatly reduce the number of trainable parameters for this model-family (Hu et al., 2021). We used the provided training dataset for training and the validation dataset to test generalization capabilities after each epoch. In the final stage, we assessed our system's performance on the dev dataset, utilizing a hierarchical version of the F1-score metric (hF1) following (Kiritchenko et al., 2006). The full set of hyperparameters we used is shown in Table 3.

5.1 Determining Optimal Thresholds

For each sample, our model outputs one logit for each class. Thus, we need to decide on a threshold, determining the decision boundary for assigning labels to instances based on their predicted probabilities. As the commonly used threshold of 0.5

Table 1: Comparison of hF1-scores and averages across all languages of our system and other top-performing systems. The corresponding ranks are provided in brackets.

System	en	bg	md	ar	Avg
Ours	0.697 (3)	**0.568 (1)**	**0.512 (1)**	**0.476 (1)**	**0.563 (1.5)**
NLPNCHU	0.663 (6)	0.517 (3)	0.462 (5)	0.475 (2)	0.529 (4.0)
914isthebest [2]	**0.752 (1)**	0.463 (11)	0.369 (14)	0.360 (13)	0.486 (9.75)
BCAmirs [3]	0.699 (2)	0.448 (13)	0.393 (12)	0.396 (9)	0.484 (9.0)

appeared non-optimal for our task based on preliminary testing, we implemented different strategies aiming to find optimal thresholds on the validation dataset. To systematically find the best threshold, we predetermined a spectrum of threshold levels to investigate. We experimented with (i) picking the same global threshold for all classes, and (ii) optimizing the threshold for each class individually. We computed the accuracy and F1-score for each threshold-label combination and selected the best outcomes for both metrics respectively. For both variants, we output all classes with probabilities above the threshold as well as all parents of the selected nodes.

6 Results

This paragraph discusses the influence of various hyperparameters, our ranking on the leaderboard and provides a detailed error analysis.

6.1 Influence of Hyperparameters

We tested the influence of both pre-processing styles, the two variants of the loss calculation, different learning rates and the custom classification head we designed. As shown in Table 5, the pre-processing variant has a negligible impact, with all models performing almost identical for both *cleaned* and *all_lower* data. Surprisingly, all models perform worse when weighting classes based on their inverse frequency in the binary cross-entropy loss. A possible reason for this is the high imbalance of our dataset (see Section 3). Weighted loss prioritizes minimizing the loss for minority classes, potentially compromising accuracy for majority classes, leading to sub-optimal overall results. The addition of our custom classification head improves our results up to eleven percent points and two percent points on average. Strikingly, *bert-large-cased* performs the worst and models pre-trained on hateful content can outperform their foundation counterparts. While *bert-hateful-memes-expanded* achieves even better results than models with a higher parameter count, *hateBERT* performs worse than the BERT-base model. Lastly, 5×10^{-5} was the best learning rate for all models tested in the grid search. Nevertheless, first experiments with LLaMA 2 revealed, that a learning rate of 5×10^{-4} works better for this model family. Using these findings, we decided to train a LLaMA 2 13b model using the *all_lower* pre-processing style with our custom classification head, a learning rate of 5×10^{-4} and no weighted loss. The LLaMA 2 models outperform the other models with the chosen parameter selection. We further observed that global thresholds consistently yielded superior performance compared to selecting single thresholds for each class. The optimal thresholds of our experiments range between 0.2 and 0.4 and vary depending on the base model and other parameters. We assume that the inferior performance of individual thresholds stems from our methodology of including all ancestors of a predicted leaf in the output, regardless of their assigned probabilities. As a result, inaccuracies at the lowest hierarchy level disproportionately affect our system's precision due to the compounded errors in ancestor predictions.

6.2 Main Results

A total of 33 teams competed in the subtask. Table 1 compares our system against other top-performing systems across all evaluated languages using the official test results. Our framework consistently ranks among the top three across all languages, securing the top position for Bulgarian (bg), North Macedonian (md) and Arabic (ar) datasets. It achieves the highest average hierarchical F1-score and the highest average leaderboard ranking. This demonstrates the versatility of our approach, underlining our methodology's effectiveness and adaptability to non-English languages. Table 2 presents

[3](Dailin Li and Lin, 2024)

[3](Amirhossein Abaskohi and Carenini, 2024)

hierarchical performance on the dev dataset for our four distinct models trained using varied seeds, in addition to their ensemble which was used for the final submission.

The highest performing individual model records a hF1 of 0.682, while the ensemble demonstrates an enhanced score of 0.690. This indicates that leveraging the outputs from multiple independently trained models can lead to improved results. Despite similar hF1 scores across models, variations of up to four and seven percent points in hierarchical precision (hP) and hierarchical recall (hR) respectively suggest differing error patterns and strengths among the models. This disparity highlights the efficacy of our ensemble approach, showcasing its capacity to amalgamate diverse insights from the dataset.

6.3 Error Analysis

In this chapter, we will dive deeper into the shortcomings of our system regarding the performance of our ensemble model on the dev dataset (Figure 2). Overall, the distribution of labels predicted by our system closely aligns with the ground truth. However, our system exhibits a bias towards predicting classes with a larger number of samples, leading to a higher frequency of these labels in our output. Conversely, labels with fewer occurrences in the training data are underrepresented in our predictions, leading to lower F1-scores in comparison. Some leaves with very few training samples such as *Presenting Irrelevant Data (Red Herring)* (59) and *Obfuscation, Intentional vagueness, Confusion* (21) are never predicted by our system, leading to a F1-score of zero. Interestingly, despite *Appeal to authority* only occurring very rarely in the training data (850), our system achieves an F1-score of 0.892 in this class. This label describes that a claim is being stated as true simply because a valid authority or expert on the issue said it was true, without any other supporting evidence offered (Dimitrov et al., 2024). We therefore assume the label to be easier to predict than other classes, as the occurrence of certain authorities or names in particular at the end of a sentence are a strong indicator for this persuasion technique. It is noticeable, that our model is able to differentiate well at the first hierarchy level: Ethos, Pathos and Logos, achieve F1-scores of over 60%. Similar observations can be made for the non-English test datasets (Figure 3, Figure 4, Figure 5).

Table 2: Hierarchical results on the dev dataset for our four distinct models trained using various seeds and the ensemble of these four models.

System	hP	hR	hF1
1	0.623	0.745	0.679
2	0.661	0.673	0.667
3	0.643	0.698	0.669
4	0.631	0.740	0.682
Ensemble	0.636	0.754	0.690

7 Conclusion

In this paper, we introduced a robust system to classify hierarchically structured persuasion techniques in a meme-corpus for the SemEval challenge 2024 Task 4 Subtask 1. Our system achieved a top-three ranking for each language individually and outperforms every other system averaged over all languages. A key aspect of our approach is the incorporation of the label hierarchy using a custom classification head that models the individual layers of the hierarchy. This classification head can be used atop of different LLMs and improves the performance by up to 11 percent points. We employed a grid-search across various models and hyperparameters to approximate optimal parameters for a LLaMA 2 13b model that then produces the embedding for the classification head. Interestingly, weighting the loss to increase the influence of classes with fewer samples did not improve the overall performance. In addition, picking the same classification threshold for each class worked better than searching one for each label individually.

There are multiple possibilities to build upon the success of our system: First, the organizers suggested similar data sources that could be used for pre-training. Additionally, upgrading to a bigger LLM, such as LLaMA 2 70b, known for its superior performance over smaller LLaMA 2 variants, could further elevate our system's capabilities. Moreover, extending our hyperparameter tuning could uncover better model configurations. Our methodology for parent-node selection could be refined by discarding parent nodes selected by children if the ancestor itself has low confidence. Lastly, feature stacking could be used to create a powerful model that incorporates features generated by other models in its classification head.

Acknowledgements

This work is partially supported by the MOTIV research project funded by the Bavarian Research Institute for Digital Transformation (bidt), an institute of the Bavarian Academy of Sciences and Humanities. Additional resources, were provided by denkbares GmbH. The authors are responsible for the content of this publication.

References

Lele Wang Amirhossein Abaskohi, Amirhossein Dabiriaghdam and Giuseppe Carenini. 2024. Bcamirs at semeval-2024 task 4: From visuals to word: A multimodal and multilingual exploration of persuasion in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Xin Zou Junlong Wang Peng Chen Jian Wang Liang Yang Dailin Li, Chuhan Wang and Hongfei Lin. 2024. 914isthebest at semeval-2024 task 4: Cot-based data augmentation strategy for persuasion techniques detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Huggingface. LlamaForSequenceClassification. Accessed 2024-01-26.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A. Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence*, pages 395–406, Berlin, Heidelberg. Springer Berlin Heidelberg.

limjiayi. 2021. limjiayi/bert-hateful-memes-expanded. Accessed 2024-02-11.

Rundong Liu, Wenhan Liang, Weijun Luo, Yuxiang Song, He Zhang, Ruohua Xu, Yunfeng Li, and Ming Liu. 2023. Recent advances in hierarchical multi-label text classification: A survey.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor

Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Hyperparameters

Table 3: Search space for hyperparameter optimization.

Parameter	**Values**
Model	bert-base-cased, bert-base-uncased, hateBERT, bert-hateful-memes-expanded, bert-large-cased, bert-large-uncased, xlm-roberta-base, xlm-roberta-large, llama-2-7b, llama-2-13b
Batch Size	128
Epochs	10
LR	5×10^{-5}, 5×10^{-4}
Style	all_lower, cleaned
Weight Loss	True, False
Custom Head	True, False

B Grid Search results

Table 4: Results of a grid-search on the dev dataset for BERT and RoBERTa models across all hyperparameters.

Model	LR	Style	Weight Loss	Custom Head	hP	hR	hF1
bert-large-cased	5×10^{-5}	cleaned	True	True	0.429	0.718	0.537
bert-large-cased	5×10^{-5}	cleaned	True	False	0.488	0.651	0.558
bert-large-cased	5×10^{-5}	all_lower	True	True	0.436	0.705	0.539
bert-large-cased	5×10^{-5}	all_lower	True	False	0.450	0.722	0.554
bert-large-cased	5×10^{-5}	cleaned	False	True	0.600	0.612	0.606
bert-large-cased	5×10^{-5}	cleaned	False	False	0.540	0.638	0.585
bert-large-cased	5×10^{-5}	all_lower	False	True	0.589	0.614	0.601
bert-large-cased	5×10^{-5}	all_lower	False	False	0.544	0.689	**0.608**
hateBERT	5×10^{-5}	cleaned	True	True	0.423	0.742	0.539
hateBERT	5×10^{-5}	cleaned	True	False	0.469	0.634	0.539
hateBERT	5×10^{-5}	all_lower	True	True	0.477	0.651	0.550
hateBERT	5×10^{-5}	all_lower	True	False	0.420	0.732	0.534
hateBERT	5×10^{-5}	cleaned	False	True	0.572	0.651	**0.609**
hateBERT	5×10^{-5}	cleaned	False	False	0.551	0.661	0.601
hateBERT	5×10^{-5}	all_lower	False	True	0.549	0.669	0.603
hateBERT	5×10^{-5}	all_lower	False	False	0.553	0.661	0.602
bert-base-cased	5×10^{-5}	cleaned	True	True	0.449	0.717	0.552
bert-base-cased	5×10^{-5}	cleaned	True	False	0.477	0.624	0.541
bert-base-cased	5×10^{-5}	all_lower	True	True	0.478	0.691	0.565
bert-base-cased	5×10^{-5}	all_lower	True	False	0.483	0.614	0.541
bert-base-cased	5×10^{-5}	cleaned	False	True	0.520	0.693	0.594
bert-base-cased	5×10^{-5}	cleaned	False	False	0.510	0.654	0.573
bert-base-cased	5×10^{-5}	all_lower	False	True	0.567	0.665	**0.612**
bert-base-cased	5×10^{-5}	all_lower	False	False	0.533	0.674	0.595
bert-base-uncased	5×10^{-5}	cleaned	True	True	0.458	0.723	0.561
bert-base-uncased	5×10^{-5}	cleaned	True	False	0.417	0.737	0.532
bert-base-uncased	5×10^{-5}	all_lower	True	True	0.490	0.664	0.564
bert-base-uncased	5×10^{-5}	all_lower	True	False	0.426	0.721	0.535
bert-base-uncased	5×10^{-5}	cleaned	False	True	0.579	0.659	**0.616**
bert-base-uncased	5×10^{-5}	cleaned	False	False	0.549	0.633	0.588
bert-base-uncased	5×10^{-5}	all_lower	False	True	0.571	0.662	0.613
bert-base-uncased	5×10^{-5}	all_lower	False	False	0.551	0.662	0.601

Table 5: Results of a grid-search on the dev dataset for BERT and RoBERTa models across all hyperparameters. Additionally, the outcomes of LLaMA 2-models for the approximated best configurations are shown.

Model	LR	Style	Weight Loss	Custom Head	hP	hR	hF1
xlm-roberta-base	5×10^{-5}	cleaned	True	True	0.455	0.715	0.556
xlm-roberta-base	5×10^{-5}	cleaned	True	False	0.461	0.659	0.543
xlm-roberta-base	5×10^{-5}	all_lower	True	True	0.449	0.707	0.550
xlm-roberta-base	5×10^{-5}	all_lower	True	False	0.446	0.652	0.530
xlm-roberta-base	5×10^{-5}	cleaned	False	True	0.561	0.688	**0.618**
xlm-roberta-base	5×10^{-5}	cleaned	False	False	0.507	0.647	0.568
xlm-roberta-base	5×10^{-5}	all_lower	False	True	0.598	0.633	0.616
xlm-roberta-base	5×10^{-5}	all_lower	False	False	0.495	0.650	0.562
bert-large-uncased	5×10^{-5}	cleaned	True	True	0.512	0.692	0.589
bert-large-uncased	5×10^{-5}	cleaned	True	False	0.480	0.676	0.561
bert-large-uncased	5×10^{-5}	all_lower	True	True	0.508	0.723	0.596
bert-large-uncased	5×10^{-5}	all_lower	True	False	0.479	0.643	0.594
bert-large-uncased	5×10^{-5}	cleaned	False	True	0.578	0.692	0.630
bert-large-uncased	5×10^{-5}	cleaned	False	False	0.412	0.686	0.515
bert-large-uncased	5×10^{-5}	all_lower	False	True	0.608	0.654	**0.630**
bert-large-uncased	5×10^{-5}	all_lower	False	False	0.594	0.621	0.607
bert-hateful-memes-expanded	5×10^{-5}	cleaned	True	True	0.494	0.673	0.570
bert-hateful-memes-expanded	5×10^{-5}	cleaned	True	False	0.472	0.638	0.542
bert-hateful-memes-expanded	5×10^{-5}	all_lower	True	True	0.502	0.666	0.573
bert-hateful-memes-expanded	5×10^{-5}	all_lower	True	False	0.473	0.643	0.545
bert-hateful-memes-expanded	5×10^{-5}	cleaned	False	True	0.591	0.679	**0.632**
bert-hateful-memes-expanded	5×10^{-5}	cleaned	False	False	0.564	0.657	0.607
bert-hateful-memes-expanded	5×10^{-5}	all_lower	False	True	0.601	0.660	0.629
bert-hateful-memes-expanded	5×10^{-5}	all_lower	False	False	0.562	0.664	0.609
xlm-roberta-large	5×10^{-5}	cleaned	True	True	0.480	0.662	0.557
xlm-roberta-large	5×10^{-5}	cleaned	True	False	0.499	0.662	0.569
xlm-roberta-large	5×10^{-5}	all_lower	True	True	0.514	0.623	0.564
xlm-roberta-large	5×10^{-5}	all_lower	True	False	0.494	0.638	0.557
xlm-roberta-large	5×10^{-5}	cleaned	False	True	0.662	0.639	0.650
xlm-roberta-large	5×10^{-5}	cleaned	False	False	0.574	0.673	0.619
xlm-roberta-large	5×10^{-5}	all_lower	False	True	0.631	0.697	**0.662**
xlm-roberta-large	5×10^{-5}	all_lower	False	False	0.581	0.688	0.630
llama7b	5×10^{-4}	all_lower	False	True	0.648	0.684	**0.666**
llama13b	5×10^{-4}	all_lower	False	True	0.623	0.745	**0.679**

C Label distribution and F1

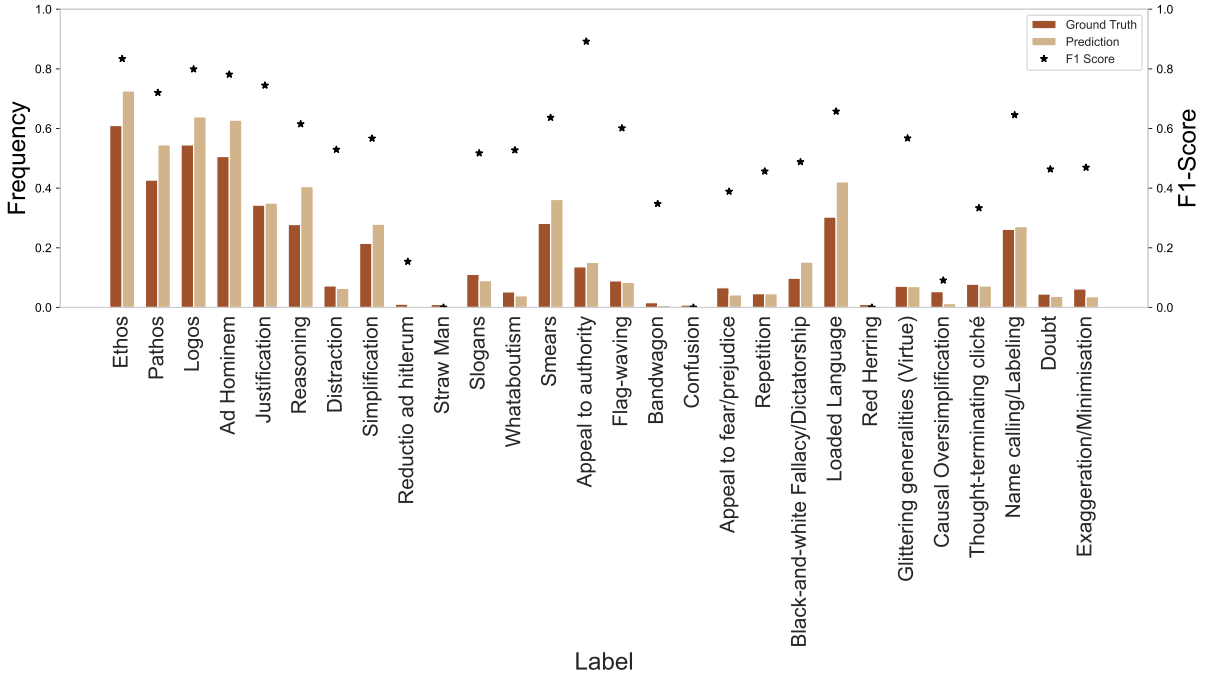


Figure 2: Distribution of labels in the **English** dev set and our system's predictions, normalized by the number of samples. The star ($\star$) indicates the F1-Score of our system for the given label.

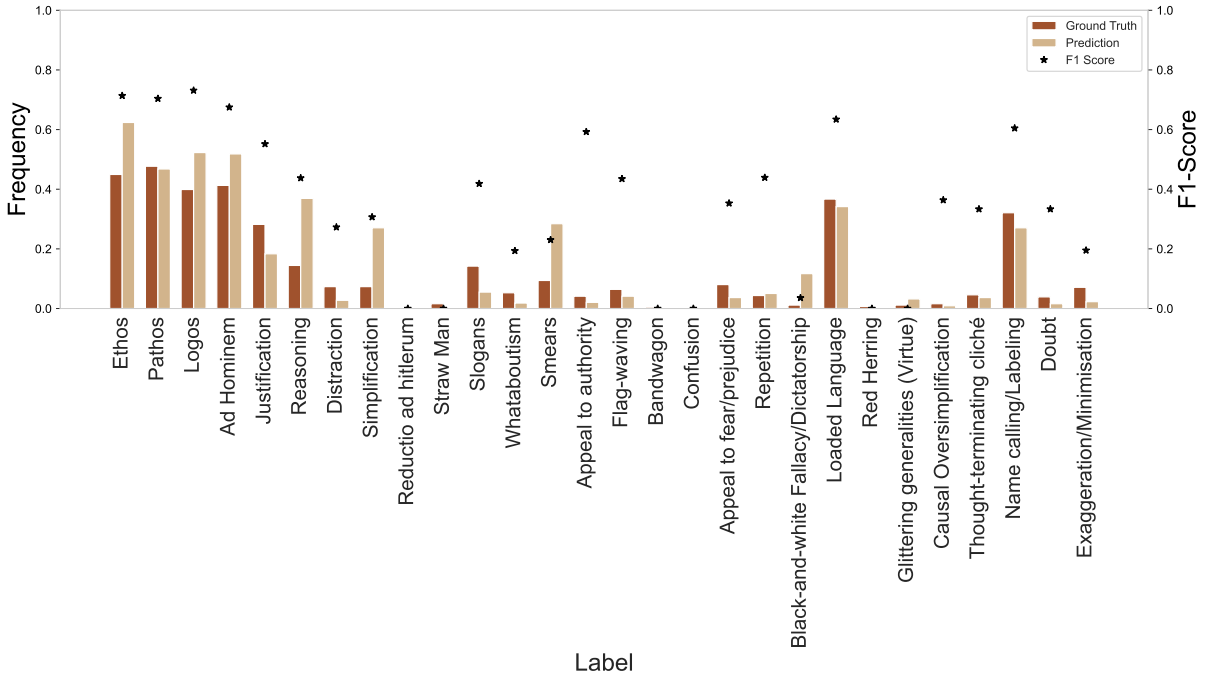


Figure 3: Distribution of labels in the **Bulgarian** test set and our system's predictions, normalized by the number of samples. The star ($\star$) indicates the F1-Score of our system for the given label.

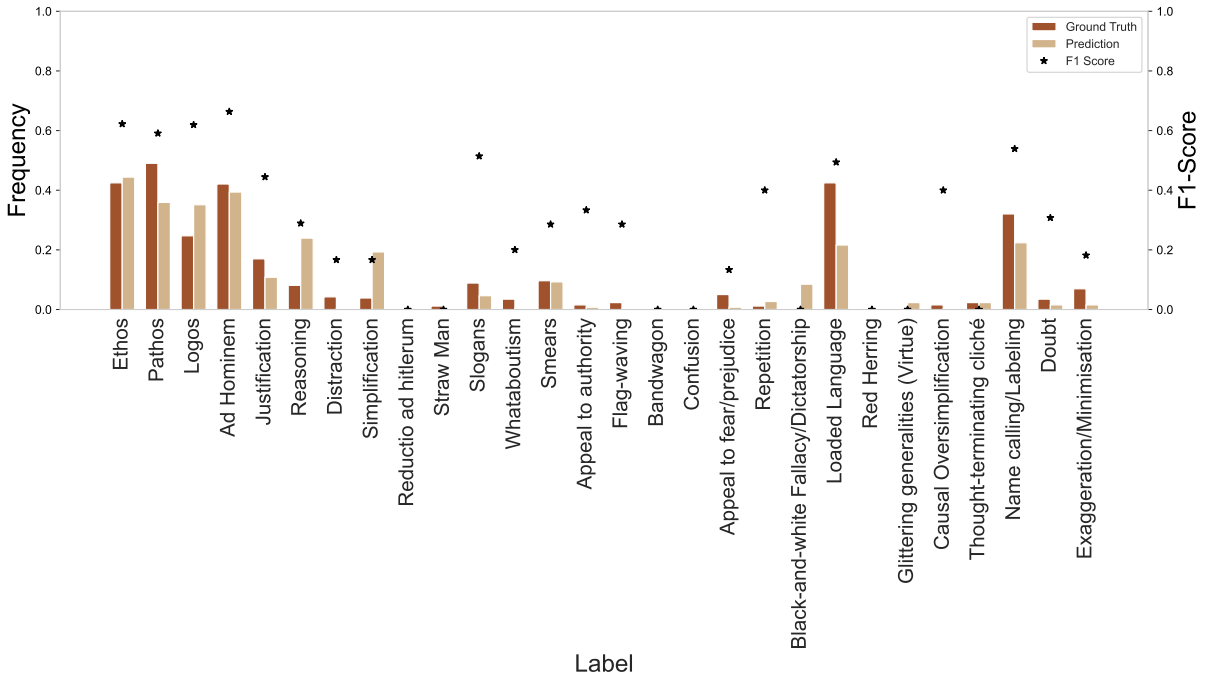


Figure 4: Distribution of labels in the **North Macedonian** test set and our system's predictions, normalized by the number of samples. The star ($\star$) indicates the F1-Score of our system for the given label.

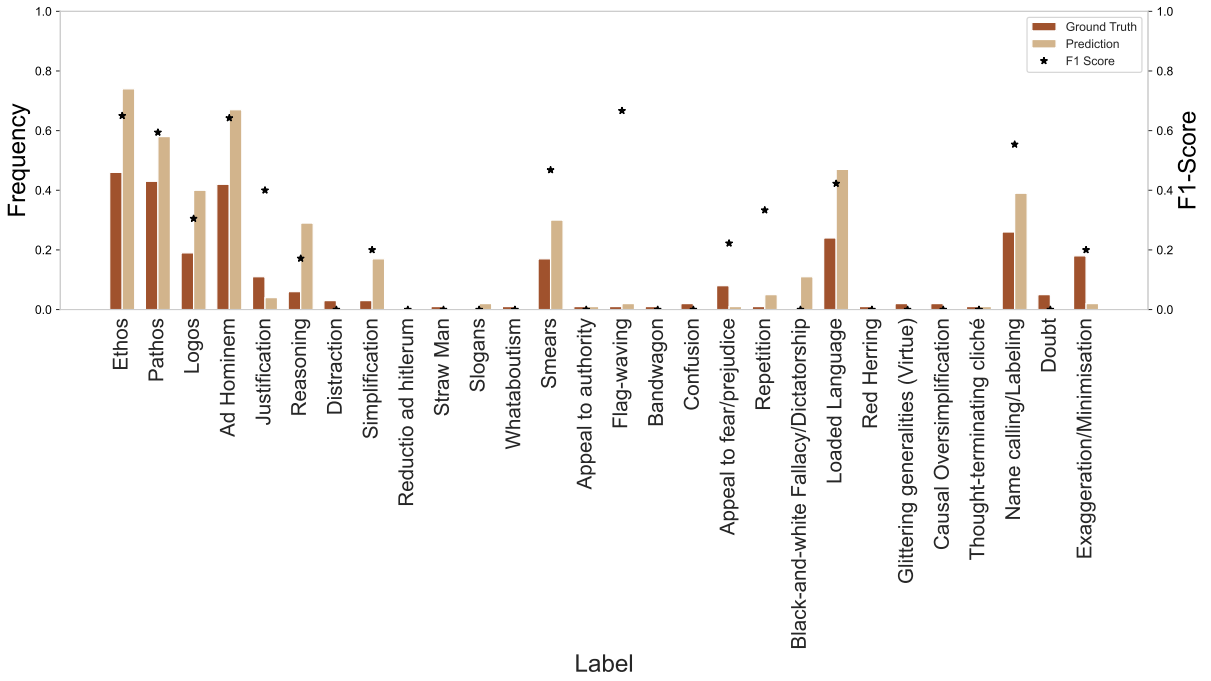


Figure 5: Distribution of labels in the **Arabic** test set and our system's predictions, normalized by the number of samples. The star ($\star$) indicates the F1-Score of our system for the given label.

D-NLP at SemEval-2024 Task 2: Evaluating Clinical Inference Capabilities of Large Language Models

Duygu Altinok
Deepgram Research, USA
duygu.altinok@deepgram.com

Abstract

Large language models (LLMs) have garnered significant attention and widespread usage due to their impressive performance in various tasks. However, they are not without their own set of challenges, including issues such as hallucinations, factual inconsistencies, and limitations in numerical-quantitative reasoning. Evaluating LLMs in miscellaneous reasoning tasks remains an active area of research. Prior to the breakthrough of LLMs, Transformers had already proven successful in the medical domain, effectively employed for various natural language understanding (NLU) tasks. Following this trend, LLMs have also been trained and utilized in the medical domain, raising concerns regarding factual accuracy, adherence to safety protocols, and inherent limitations. In this paper, we focus on evaluating the natural language inference capabilities of popular open-source and closed-source LLMs using clinical trial reports as the dataset. We present the performance results of each LLM and further analyze their performance on a development set, particularly focusing on challenging instances that involve medical abbreviations and require numerical-quantitative reasoning. Gemini, our leading LLM, achieved a test set F1-score of 0.748, securing the ninth position on the task scoreboard. Our work is the first of its kind, offering a thorough examination of the inference capabilities of LLMs within the medical domain.

1 Introduction

Large language models (LLMs) have brought about a paradigm shift in the field of Natural Language Processing (NLP) (Kojima et al., 2023; Wei et al., 2022). Their exceptional performance across various tasks has led to a surge in real-world applications utilizing LLM-based technology. However, a notable drawback of LLMs is their propensity to generate plausible yet incorrect information, commonly referred to as "hallucinations" (Ji et al., 2023).

The remarkable breakthrough of LLMs has raised questions regarding their "intelligent" capabilities, particularly in reasoning and inference (Zhao et al., 2023; Chang et al., 2023; Laskar et al., 2023). Two specific areas that have garnered significant attention in relation to LLMs' reasoning abilities are numerical-quantitative reasoning and natural language inference. These areas are considered integral to human intelligence, prompting researchers to establish benchmarks and evaluate LLM performance in these domains (Stolfo et al., 2023; Yuan et al., 2023). LLMs often exhibit limited performance in solving arithmetic reasoning tasks, frequently producing incorrect answers (Imani et al., 2023). Unlike natural language understanding, math problems typically possess a single correct solution, making the accurate generation of solutions more challenging for LLMs. Regarding NLI, performance reduction can be observed due to shortcut learning (Du et al., 2023) and hallucinations (McKenna et al., 2023). These investigations aim to discern whether LLMs are mere memorizers of training data or possess genuine logical reasoning abilities.

The volume of medical publications, including clinical trial data, has experienced a significant upsurge in recent years. The SemEval-2023 Task 7, known as Multi-Evidence Natural Language Inference for Clinical Trial Data (NLI4CT), aimed to address the challenge of large-scale interpretability and evidence retrieval from breast cancer clinical trial reports (Jullien et al., 2023). This task required multi-hop biomedical and numerical reasoning, which are crucial for developing systems capable of interpreting and retrieving medical evidence on a large scale, thereby facilitating personalized evidence-based care. While the previous iteration of NLI4CT resulted in the development of LLM-based models (Zhou et al., 2023;

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 613–627

Kanakarajan and Sankarasubbu, 2023; Vladika and Matthes, 2023) achieving high performance (e.g., F1-score $\approx 85\%$), the application of LLMs in critical domains, such as real-world clinical trials, necessitates further investigation. Consequently, the second iteration of NLI4CT, SemEval-2024 Task 2, titled "Safe Biomedical Natural Language Inference for Clinical Trials" (Jullien et al., 2024) is proposed, featuring an enriched dataset that includes a novel contrast set obtained through interventions applied to statements in the NLI4CT test set. Our work involves the evaluation of various popular open-source and closed-source LLMs on the development and test sets to explore their reasoning capabilities in the domain of medical NLI. We present the results by thoroughly analyzing the performance on the development set, with the best-performing LLM ranking ninth on the task leaderboard. We have made the results on the development set available on our GitHub repository[1].

Another aspect of our work was that we deliberately refrained from investing significant effort into prompting or experimenting with different prompts. Additionally, we aimed to showcase the remarkable development of LLMs, demonstrating their capacity to effectively engage with the task while minimizing dependence on the prompt.

2 Related Work

With the emergence of large language models (LLMs), there has been a growing interest in exploring their capabilities within the clinical domain. Recent studies have delved into both the potential of LLMs and the associated risks when applied in clinical settings. For instance, (Hung et al., 2023) conducted experiments utilizing GPT-3.5 on various medical NLP datasets, assessing metrics such as factuality and safety, ultimately highlighting the high level of safety offered by GPT-3.5 [2]. (Pal et al., 2023) focused on the challenges posed by hallucinations in LLMs and proposed a benchmark dataset called Med-HALT (Medical Domain Hallucination Test) to evaluate popular LLMs on this front.

Regarding the reasoning capabilities of LLMs, (Kwon et al., 2024) introduced a diagnostic framework that prioritizes reasoning and employs prompt-based learning. The study specifically focused on clinical reasoning for disease diagnosis, where the LLMs generate diagnostic rationales to provide insights into patient data and the reasoning path leading to the diagnosis, known as Clinical Chain-of-Thought (Clinical CoT), using GPT-3.5 and GPT-4 (OpenAI, 2024). Notably, none of the previous studies simultaneously examined the performance of both open-source and closed-source LLMs, particularly with a comprehensive focus on inference. Consequently, our work stands as the first of its kind in this regard.

3 Task and Dataset Description

The clinical trials used to construct the dataset were sourced from ClinicalTrials.gov[3], a comprehensive database managed by the U.S. National Library of Medicine. ClinicalTrials.gov contains information on various clinical studies conducted worldwide, both publicly and privately funded. The dataset specifically focuses on clinical trials related to breast cancer and includes a total of 1,000 trials written in English.

- **Eligibility Criteria:** This includes a set of conditions that determine the eligibility of patients to participate in the clinical trial. These criteria may include factors such as age, gender, and medical history.
- **Intervention:** This field provides information about the type, dosage, frequency, and duration of treatments being studied within the clinical trial.
- **Results**: The results section of each CTR reports the outcome of the trial, including data such as the number of participants, outcome measures, units of measurement, and the observed results.
- **Adverse Events**: This field documents any unwanted side effects, signs, or symptoms observed in patients during the course of the clinical trial.

For the task at hand, each CTR may contain one or two patient groups, known as cohorts or arms, which may receive different treatments or have different baseline characteristics.

The dataset consists of a total of 7,400 statements. These statements were divided into a training dataset comprising 1,700 statements, a development dataset containing 200 statements, and a

[1] https://github.com/DuyguA/SemEval2024_NLI4CT

[2] https://platform.openai.com/docs/models/gpt-3-5

[3] https://clinicaltrials.gov

Model	Release Date	Params
GPT-3.5	Mar-2022	x
Claude	Mar-2023	x
Gemini Pro	Dec-2023	x
PaLM	Mar-2023	540B
Falcon 40B	May-2023	40B
Mixtral 8x7B	Dec-2023	12B
Llama 2 70B	Jul-2023	130GB

Table 1: Comparison of the LLMs used in our work, indicating the parameter sizes for known closed-source LLMs and denoting unknown parameter sizes with "x".

Model	Acc	F1	Prec	Recall
Gemini Pro	0.82	0.81	0.82	0.8
Claude	0.81	0.80	0.81	0.81
PaLM	0.79	0.78	0.79	0.79
Falcon 40B	0.745	0.74	0.74	0.74
GPT-3.5	0.705	0.7	0.711	0.70
Llama 2 70B	0.675	0.67	0.68	0.67
Mixtral 8x7B	0.655	0.64	0.67	0.65

Table 2: Accuracy, macro F1-score, precision and recall results on the development set for each LLM.

Model	F1	Faith	Consist
Gemini Pro	0.75	0.83	0.74
Claude	0.73	0.83	0.72
PaLM	0.72	0.87	0.73
Falcon 40B	0.702	0.569	0.609
GPT-3.5	0.684	0.74	0.66
Llama 2 70B	0.682	0.693	0.638
Mixtral 8x7B	0.604	0.899	0.73

Table 3: Macro F1-score, faithfulness and consistency results on the test set for each LLM.

hidden test dataset consisting of 5,500 statements. The statements can be categorized into two types: those that are solely related to a single CTR and others that involve a comparison between two different reports. Each statement in the dataset is labeled as either "entailment" or "contradiction". Figure 1 shows an example statement from the training set.

The task primarily involves binary classification, aiming to predict whether the label corresponds to entailment or contradiction. The evaluation process encompasses three aspects. Initially, the macro F1-score is computed based on the binary classification results. Subsequently, two semantic evaluations are conducted: faithfulness and consistency. Faithfulness assesses the system's ability to arrive at accurate predictions for the correct reasons, while consistency measures the system's ability to produce consistent outputs for semantically equivalent problems. The task organizers evaluate faithfulness by providing semantically altered instances, and consistency by providing preserved instances for comparison.

4 Language Model Performance Evaluation

This section aims to provide a detailed analysis of the performance achieved by each individual LLM. Based on the evaluation of various LLMs, including closed-source models like GPT-3.5 (ChatGPT), Claude (Anthropic, 2023), and Gemini Pro (Gemini Team, 2023), as well as open-source models like Falcon 40B (Almazrouei et al., 2023), Mixtral 8x7B (Jiang et al., 2024), and Llama 2 70B (Touvron et al., 2023), the performance of these models was assessed on the dev and test sets. Table 1 provides comprehensive information regarding the release dates and parameter sizes, measured in token size, for each LLM.

All conversations took place on the Poe.com platform, providing users with a seamless chat experience. To transmit both the development set and the test set instances, we utilized an API wrapper code in a Python script, which can be accessed in our GitHub repository. As mentioned earlier, we intentionally avoided extensive prompting and instead employed a straightforward, consistent prompt for all instances. Each model's chat session commenced with a greeting, followed by a brief introductory sentence regarding the task, and subsequently, all instances were dispatched via the Python script. Appendix A provides information regarding the prompts.

Table 2 and Table 3 presents a concise overview of the results obtained on the dev and test sets. Gemini Pro emerged as the best-performing model, ranking first on both the dev and test sets. Following Gemini Pro, Claude and PaLM, two closed-source LLMs, secured the second and third positions, respectively. Falcon 40B, an open-source LLM, achieved the fourth place and outperformed GPT-3.5. The last two positions were occupied by two open-source LLMs, Llama 2 70B and Mixtral 8x7B.

In the next section, we delve into the detailed performance analysis of the language models on the development set, focusing on specific cases of

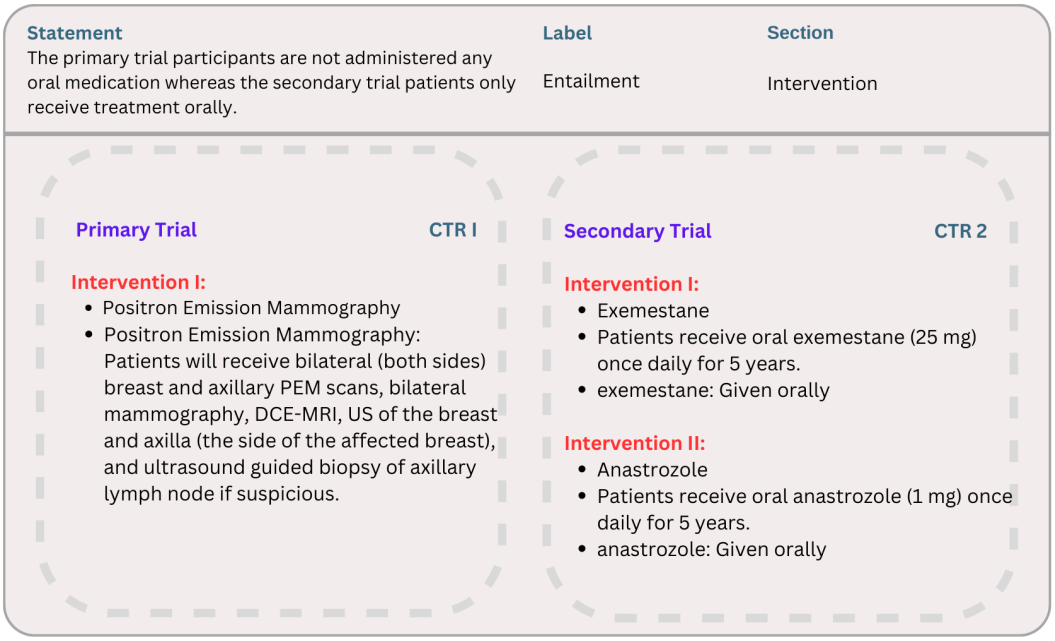


Figure 1: An example comparison task from the training set with two CTRs.

Model	Incorrect
Gemini Pro	36
Claude	38
PaLM	42
Falcon 40B	51
GPT-3.5	59
Llama 2 70B	65
Mixtral 8x7B	69

Table 4: Number of incorrect predictions on the development set of 200 instances for each LLM.

interest.

4.1 General Performance Evaluation

Among the top-ranking LLMs, namely Gemini Pro, Claude, PaLM, and Falcon 40B, their performance on the development set was indeed remarkable. The number of inaccurate predictions made by each LLM on the development set of 200 instances is presented in Table 4. There were only 3 instances in the development set that were incorrectly predicted by all LLMs.

Among the top-performing LLMs, a set of 12 instances emerged as particularly challenging, denoted as "difficult instances". These instances present a significant challenge, as none of the top three performer LLMs in the set - Gemini Pro, Claude and PaLM, were able to accurately predict their outcomes. Within this subset, 2 instances involved medical abbreviations, 3 featured numerical entities, and the remaining 7 were categorized as "plain instances". The subsequent subsections will delve into a detailed analysis of the difficult instances encompassing numerical entities and abbreviations. For the present discussion, our attention will be directed towards the plain difficult instances.

One particular difficult instance is illustrated in Figure 1, where the top two performing LLMs provided the answers given by Figure 2.

Gemini made a somewhat fair mistake by incorporating knowledge from the training data, suggesting that PET includes orally administered tracer drugs. However, it should be noted that tracer drugs do not qualify as any form of medication for treating cancer or other diseases. In this case, Gemini falters by "assuming too much" and "relying too heavily on prior knowledge." It is important to clarify that this is not a hallucination. On the other hand, Claude correctly deduced that there were no oral medications in the primary trial and that oral medication was administered in the secondary trial. However, Claude failed to synthesize this information and draw a conclusion, indicating a breakdown in reasoning from evidence. Similarly, PaLM and Falcon 40B encountered the same issue as Claude. While they accurately pointed out the evidence, they failed in the reasoning process. In the case of PaLM, it did not provide any indications of en-

GEMINI
Contradiction. The statement “the primary trial participants are not administered any oral medication whereas the secondary trial patients only receive treatment orally” is a contradiction of the given trial information. The primary trial intervention section states that patients will receive Positron Emission Mammography, which involves various PEM views being performed on both the breast and axilla. This indicates that the primary trial participants are administered some form of oral medication, as the PEM views are likely to involve the use of a radioactive tracer that is taken orally. Therefore, it is not accurate to say that the primary trial participants are not administered any oral medication. Hence, the statement is a contradiction of the trial information.
CLAUDE
Contradiction. The primary trial intervention specifies: - Positron Emission Mammography scans. The secondary trial intervention specifies: - Exemestane 25mg oral daily for 5 years.- Anastrozole 1mg oral daily for 5 years. The statement claims:- Primary trial does not administer oral medications\n- Secondary trial only provides oral treatments. However, the primary trial does not mention any oral or other medications. And the secondary trial clearly describes oral aromatase inhibitors as the interventions. As the primary trial information does not validate the negative claim about oral medications, and the secondary trial positively describes oral treatments, the statement is contradicted. Therefore, the answer is contradiction - the trial intervention details conflict with the assertions made in the statement.

Figure 2: An example comparison task from the training set with two CTRs.

gaging in reasoning. Falcon 40B made a literal mistake by becoming mired in the intricacies of the language. Its answer includes the statement, "the statement cannot be directly entailed from the intervention information provided. Specifically, while the primary trial does not mention oral medication, the secondary trial does not exclusively mention oral medication, contradicting the statement."

These mistakes range from overthinking, failure to follow the evidence, inadequate reasoning, to becoming excessively focused on minute details—a clear manifestation of the inherent challenges that LLMs face. The less performing LLMs, namely GPT-3.5, LLama 2 70B, and Mixtral 7x8B, demonstrated a decent performance considering the task difficulty. However, they exhibited a relatively higher frequency of failures in reasoning and inference compared to the top-performing LLMs.

Having examined the plain difficult instances, we now turn our attention to evaluating the performance of the LLMs on instances containing medical abbreviations.

4.2 Abbreviated Instances Performance Evaluation

In our development set, we identified 31 instances that contained medical abbreviations. We used the ScispaCy package’s abbreviation detector to extract these instances.

Among the top performers, Gemini, Claude, PaLM, and Falcon 40B made 4, 6, 7, and 8 mistakes, respectively, in handling these abbreviations. The bottom performers, GPT-3.5, Llama 2 70B, and Mixtral 8x70B, made 10, 8, and 8 mistakes, respectively.

Upon closer examination, we found that all of the LLMs were able to correctly resolve the meanings of the medical abbreviations. However, they made mistakes due to other reasoning problems.

For example, none of Gemini’s four mistakes in handling abbreviations were related to resolving their meanings. Similarly, the other LLMs also failed the task primarily due to quantitative-numerical reasoning failures. Appendix B showcases a more comprehensive example of this particular type of occurrence and the corresponding failure.

Overall, the performance of all LLMs in resolving abbreviations was commendable. However, as mentioned before, the majority of failures stemmed from challenges in numerical-quantitative reasoning.

4.3 Numerical Instances Performance Evaluation

Our development set contained 78 instances with numerical entities. We employed the spaCy package (Honnibal and Montani, 2017) and its NER component to identify these entities. To ensure comprehensive semantic evaluation, we combined

ScispaCy and spaCy models.

Among the top-performing LLMs, Gemini, Claude, PaLM, and Falcon 40B made 13, 14, 18, and 19 mistakes, respectively. Notably, the bottom performers, GPT-3.5, Llama 2 70B, and Mixtral 8x70B, made significantly more mistakes (21, 26, and 24, respectively).

Interestingly, the top performers, Gemini and Claude, made the same mistakes on numerical instances in the development set. Upon examining their responses, we observed that they performed arithmetic operations and reasoned based on the calculated results. In Appendix B, Figure 8 and Figure 9 portray instances of successful outcomes achieved by our LLMs. These figures demonstrate accurate performance in arithmetic operations and logical deduction.

However, even the top performers made occasional errors. For instance, Gemini provided an incorrect answer where there was no evidence of arithmetic operations or reasoning: "The primary trial adverse events section shows that there were 10 patients in cohort1 who suffered adverse events out of a total of 67 patients. Therefore, it is accurate to say that over 1/6 patients in cohort1 of the primary trial suffered adverse events." This suggests that Gemini did not calculate 1/6 of the total number of patients (67).

Among the numerical instances incorrectly predicted by Gemini and Claude, we found no instances where arithmetic calculations were performed. Conversely, correctly predicted instances, such as the one shown in Figure 13 in Appendix B, involved at least one mathematical operation that was logically connected to the rest of the argument. We determined these logical connections by analyzing the dependency tree of the answers, as explained in Appendix B. Our findings indicate that when LLMs demonstrate signs of performing arithmetic operations, their results are generally reliable. Conversely, when there is no evidence of arithmetic operations, the result is likely incorrect.

The other top performers, PaLM and Falcon 40B, exhibited similar behavior to Gemini and Claude. They performed arithmetic operations and made deductions based on those operations. When they failed, they did not provide any numerical clues.

The bottom performer, GPT-3.5 was able to perform arithmetic operations. However, it struggled with simple quantity comparisons, such as n<m for random integers. Mixtral 8x7B also faced similar challenges.

Llama 2 70B performed particularly poorly on numerical instances. For the example in Figure 9, where other LLMs succeeded by performing arithmetic operations, Llama failed completely. It provided an incorrect answer without any evidence of subtraction or comparison. In fact, Llama generally struggled with numerical examples, succeeding primarily in quantitative comparisons where operands were provided directly in the context without requiring mathematical processing.

In conclusion, while other LLMs demonstrated proficiency in handling numerical entities, Llama 2 70B failed to meet expectations.

5 Conclusion

Our detailed analysis of LLMs' performance on various reasoning tasks in the medical domain reveals that they are not merely passive memorizers. They possess the ability to perform numerical-quantitative reasoning, general reasoning, and abbreviation resolution, even in a highly specialized domain with unique vocabulary. Notably, Falcon 40B, an open-source LLM, demonstrated impressive performance, rivaling top closed-source LLMs.

Despite their successes, LLMs are not without limitations. Occasional nonsensical predictions highlight the need for caution when using them in high-stakes domains such as medicine. However, the results of our study are highly promising and suggest that with increased training data and computational power, LLMs have the potential to become invaluable tools in the medical field.

The future of LLMs in medicine holds exciting possibilities. As these models continue to evolve, we anticipate that they will play an increasingly significant role in healthcare, transforming the way we diagnose, treat, and prevent diseases.

6 Limitations

As mentioned in earlier sections, we utilized the Poe platform for interacting with LLMs. All the work was accomplished within the confines of a monthly subscription fee of $20. The results of GPT-4 are not included in this study due to the messaging limit imposed by the platform, which was exceeded by the number of instances in the test set.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Anthropic. 2023. Introducing claude.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Commun. ACM*, 67(1):110–120.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Chia-Chien Hung, Wiem Ben Rim, Lindsay Frost, Lars Bruckner, and Carolin Lawrence. 2023. Walking a tightrope – evaluating large language models in high-risk domains.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama AI research at SemEval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003, Toronto, Canada. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Taeyoon Kwon, Kai Tzu iunn Ong, Dongjin Kang, Seungjun Moon, Jeong Ryong Lee, Dosik Hwang, Yongsik Sim, Beomseok Sohn, Dongha Lee, and Jinyoung Yeo. 2024. Large language models are clinical reasoners: Reasoning-aware diagnosis framework with prompt-generated rationales.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks.

OpenAI. 2024. Gpt-4 technical report.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. Med-halt: Medical domain hallucination test for large language models.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schölkopf, and Mrinmaya Sachan. 2023. A causal framework to quantify the robustness of mathematical reasoning with language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten,

Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Juraj Vladika and Florian Matthes. 2023. Sebis at SemEval-2023 task 7: A joint system for natural language inference and evidence retrieval from clinical trial reports. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1863–1870, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. 2023. How well do large language models perform in arithmetic tasks?

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. THiFLY research at SemEval-2023 task 7: A multi-granularity system for CTR-based textual entailment and evidence retrieval. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1681–1690, Toronto, Canada. Association for Computational Linguistics.

A Prompts

We employed two prompts, the prompts for the individual task and comparison tasks outlined as follows:

"Below find section_name section of the primary trial of a clinical trial. Infer if the following statement entails from the given trial information. Answer should be either entailment or contradiction. Please justify the answer based on numbers. PRIMARY TRIAL section_name: trial_value STATEMENT: statement"

"Below find section_name sections of a primary trial and a secondary trial belonging to same clinical trial. Infer if the following statement entails from the given trial information. Answer should be either entailment or contradiction. Please justify the answer based on numbers. PRIMARY TRIAL section_name: trial_value1 SECONDARY TRIAL section_name: trial_value2 STATEMENT: statement".

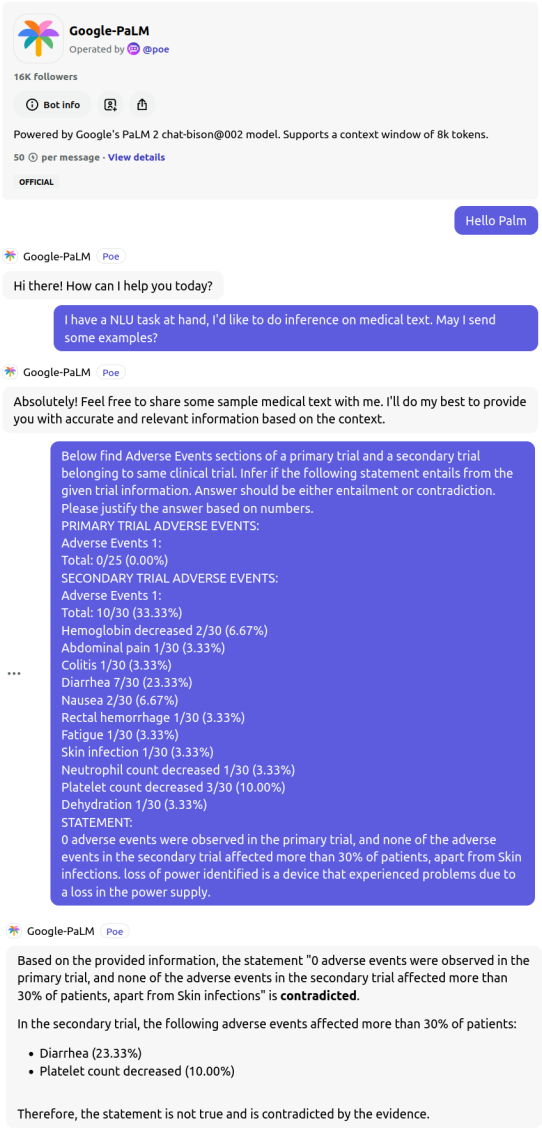


Figure 3: Initiation of the conversation with PaLM.

Figure 3 illustrates the initiation of a chat with PaLM and the method by which instances are provided during the conversation. As evident in the interaction, we maintained minimal prompting and limited additional interactions.

B Example Instances

In this section of the appendix, we present specific instances from the development set to provide readers with a concrete understanding of the performance of LLMs. Firstly, we present a challenging instance, which none of the LLMs in our study were able to correctly predict. Figure 4 depicts

this instance, which involves making an inference about the results section of a single CTR. The inference relates to PFS, a time range spanning from 7.0 to 9.9 months, with an average of 8.4 months. Consequently, the statement presents an entailment. Surprisingly, all the LLMs failed to address this instance. As depicted in Figure 5 and 6, the LLMs struggled to calculate the difference due to various reasons, such as difficulties in numerical deduction or becoming overly focused on linguistic details.

It is worth noting that this instance also includes an abbreviation, PFS, which is fully explained in the body of the CTR. Despite the LLMs demonstrating some understanding of this abbreviation, they ultimately failed due to their inability to perform the necessary numerical inference.

Subsequently, we present a numerical case study depicted in Figure 7 to showcase the numerical reasoning capabilities of the Language and Logic Models (LLMs). Impressively, almost all LLMs accurately predicted this particular case. However, Llama 2 70B exhibited a complete failure, displaying no signs of any numerical inference whatsoever. Figures 8 and 9 illustrate how other LLMs meticulously explained their reasoning step by step. They initiated the process by performing the subtraction $89\% - 88\% = 1\%$ and subsequently compared the result to the claimed amount of 13.2%.

To process numerical instances, we adopted the following approach: firstly, we utilized spaCy's Matcher component to extract all numerical expressions [4]. This component, being part of the pretrained spaCy pipelines, is incredibly helpful in extracting expressions based on patterns. These patterns can involve characteristics such as token shape, POS tags, and even entity types if the token forms part of an entity. By leveraging spaCy's built-in NER component, we could extract various numerical entity types, including cardinal numbers, ordinal numbers, percentages, and quantities. We formulated two general Matcher patterns, namely *NUMERIC OP NUMERIC* and *NUMERIC OP NUMERIC = NUMERIC*, and then generated all possible combinations of numerical entities and mathematical expressions by taking the cross product between numeric entity types and mathematical operator tokens. This comprehensive approach facilitated the extraction of all numerical expressions from the LLM answers. For identifying medical entities, we utilized the ScispaCy package, as medical entities are not included in spaCy's general-purpose NER models.

Following this, we parsed the dependency tree of the answer and determined the syntactic head of the numerical expression. We then examined whether the numerical expression attached meaningfully to the rest of the answer. For a detailed explanation of the reasoning process, refer to Figure 13.

Moving on to our list of examples, we encounter an intriguing case worth mentioning. Figure 10 presents a CTR with an empty adverse events section, making it a particularly interesting example. As depicted in Figures 11 and 12, all LLMs, except for Falcon 40B, demonstrate impressive intelligence by correctly interpreting "0/0" as indicating the absence of any adverse events, thereby resulting in an empty adverse events section. This example highlights the remarkable general language understanding and common sense reasoning abilities of LLMs, transcending the boundaries of the medical domain.

[4] https://spacy.io/api/matcher

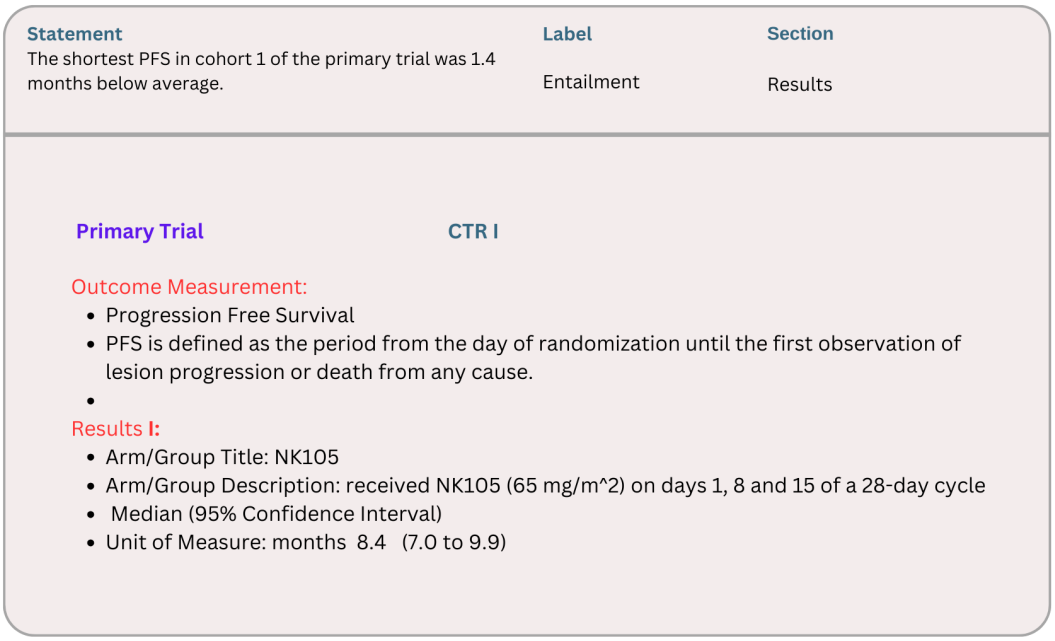


Figure 4: A challenging instance that was incorrectly predicted by the top-performing LLMs.

GEMINI
Contradiction. The statement "The shortest PFS in cohort 1 of the primary trial was 1.4 months below average" is a contradiction of the given trial information. The primary trial results section shows that the median PFS in cohort 1 was 8.4 months (95% CI: 7.0 to 9.9 months). Therefore, it is not accurate to say that the shortest PFS in cohort 1 of the primary trial was 1.4 months below average. Hence, the statement is a contradiction of the trial information.
CLAUDE
Contradiction. The primary trial results reported: - Outcome Measurement: Progression Free Survival (PFS) - Arm 1 Participants Analyzed: 211 - Arm 1 PFS Median: 8.4 months (7.0 to 9.9) - Arm 2 Participants Analyzed: 211 - Arm 2 PFS Median: 8.5 months (6.9 to 11.5). The statement claims the shortest PFS in cohort 1 was 1.4 months below average. However, the results reported PFS as medians with confidence intervals, not averages. There is no information provided about specific individual PFS durations. Since the results do not include data to validate the claim of any patient being 1.4 months below average, this contradicts the statement. Therefore, the trial results do not substantiate the assertion made in the statement. The answer is contradiction, as the reported outcomes conflict with the statement.
PALM
Contradiction. The text states that "Median (95% Confidence Interval) 8.4 (7.0 to 9.9)" in the primary trial results 1 and "Median (95% Confidence Interval) 8.5 (6.9 to 11.5)" in the primary trial results 2. This means that the shortest PFS in cohort 1 of the primary trial was not 1.4 months below average.

Figure 5: Responses of the top-performing LLMs to the selected challenging instance, where all models failed to exhibit any signs of numerical inference.

FALCON 40B
Median PFS reported is 8.4 months with a 95% CI of 7.0-9.9 months. Results 2: - Paclitaxel arm analyzed 211 participants- Median PFS reported is 8.5 months with a 95% CI of 6.9-11.5 months. Based on the numbers directly reported: - No cohort or arm assignments are specified - Median values reported are 8.4 and 8.5 months - No PFS of 1.4 months below any average is stated. Therefore, upon my precise numerical analysis, the statement cannot be logically entailed from the PFS results explicitly shared and appears to contradict them.
GPT-3.5
The statement "The shortest PFS in cohort 1 of the primary trial was 1.4 months below average" contradicts the given trial information. Justification: Based on the given trial information, the median PFS for cohort 1 (NK105 arm) is reported as 8.4 months (with a confidence interval of 7.0 to 9.9), while the median PFS for cohort 2 (Paclitaxel arm) is reported as 8.5 months (with a confidence interval of 6.9 to 11.5). Neither of these results indicates the shortest PFS being 1.4 months below the average. Therefore, the statement contradicts the given trial information as it incorrectly suggests that the shortest PFS in cohort 1 was 1.4 months below the average.
LLAMA 2 70B
Contradiction. The statement "The shortest PFS in cohort 1 of the primary trial was 1.4 months below average" implies that the shortest PFS in cohort 1 was 1.4 months less than the average PFS in cohort 1. However, the results provided in the PRIMARY TRIAL RESULTS section show that the median PFS for cohort 1 (NK105) was 8.4 months, and the median PFS for cohort 2 (Paclitaxel) was 8.5 months. Since the median PFS for both cohorts is above 8 months, it is not possible for the shortest PFS in cohort 1 to be 1.4 months below average.
MIXTRAL 8x7B
The statement The shortest PFS in cohort 1 of the primary trial was 1.4 months below average" cannot be inferred from the given trial information. The results show that the median PFS for Arm/Group Title: NK105 was 8.4 (7.0 to 9.9) and for Arm/Group Title: Paclitaxel was 8.5 (6.9 to 11.5). Neither of these results indicate that the shortest PFS was 1.4 months below average.

Figure 6: Responses of the low-performing LLMs to the selected challenging instance, which were not significantly different from the answers provided by the top LLMs.

Figure 7: A numerical example extracted from the development set, focusing on inferring information from a single CTR's results section.

GEMINI
Contradiction. The statement "there is a **13.2%** difference between the results from the two the primary trial cohorts" is a contradiction of the trial information provided. To calculate the difference between the results from the two primary trial cohorts, we can subtract the event-free survival rate in the exemestane arm from the event-free survival rate in the anastrozole arm: **89% - 88% = 1%**. Therefore, the difference between the results from the two primary trial cohorts is **1%**, not **13.2%**. Hence, the statement is a contradiction of the trial information.

CLAUDE
Contradiction. The primary trial results provide event-free survival percentages of 87-89% for the exemestane cohort and 88-90% for the anastrozole cohort. These ranges show overlap between the results of the two cohorts. There is no information given that directly states or implies a 13.2% difference.

PALM
Contradiction. The text states that the event-free survival rate in the exemestane arm was **88% (87 to 89)**, while the event-free survival rate in the anastrozole arm was **89% (88 to 90)**. This means that the difference between the two arms was only **1%**, **not 13.2%**.

FALCON 40B
The statement "there is a 13.2% difference between the results from the two primary trial cohorts" would be a contradiction. The results reported a range of **87-89%** for the exemestane cohort and **88-90%** for the anastrozole cohort. This shows only a difference of **1-2%** between the lowest and highest estimates provided. **13.2%** is much greater than the actual difference observed between these two cohorts for the 5 year event-free survival endpoint.

Figure 8: Responses of the top-performing LLMs to the selected numerical instance, showcasing impressive performance in numerical inference across all models.

GPT-3.5
To determine if there is a **13.2%** difference between the results from the two primary trial cohorts, let's compare the percentages provided in the results. In Results 1 (Exemestane group), the percentage of participants analyzed is **88% (87 to 89**), and in Results 2 (Anastrozole group), the percentage of participants analyzed is **89% (88 to 90)**. The difference between these two percentages can be calculated as follows: **89% - 88% = 1%**. Based on the provided numbers, the calculated difference is **1%**, not **13.2%**. Therefore, the statement "there is a 13.2% difference between the results from the two primary trial cohorts" contradicts the information given in the primary trial results.

LLAMA 2 70B
ENTAILMENT. The statement mentions a **13.2%** difference between the results from the two primary trial cohorts, which is within the margin of error for the confidence interval provided in the results (**87-89%** for exemestane and **88-90%** for anastrozole). Therefore, the statement entails from the given trial information.

MIXTRAL 8x7B
Contradiction. The statement is not correct. The difference between the two groups is actually **1 percentage** point (**89%** for anastrozole minus **88%** for exemestane), which is a relative difference of about **1.12%** (**1/88 * 100**). This is not the same as a **13.2%** difference as stated

Figure 9: Responses of the bottom-performing LLMs to the selected numerical instance, where all models, except for Llama 2 70B, successfully performed the subtraction operation and made the corresponding numerical inference.

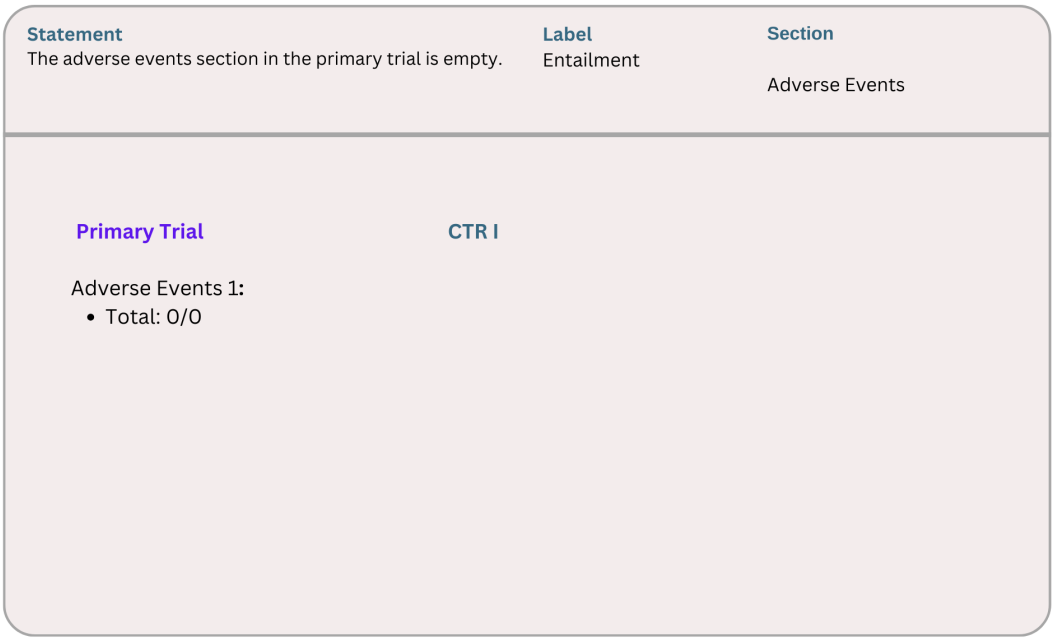


Figure 10: A CTR from the training set that is both interesting and uninteresting. It is interesting because the value of 0 denotes emptiness, but uninteresting because humans can understand 0 as representing null.

GEMINI
Entailment. The statement "The adverse events section in the primary trial is empty" is an entailment of the given trial information. The primary trial adverse events section shows a total of 0 adverse events. Therefore, it is accurate to say that the adverse events section in the primary trial is empty
CLAUDE
Entailment. The adverse events section provided reports: Adverse Events 1: Total: 0/0. The statement says "The adverse events section in the primary trial is empty". Since the only information given in the adverse events section is: Adverse Events 1: Total: 0/0. And there are no other details listed, this directly matches the statement that the adverse events section is empty. Therefore, the statement can be directly substantiated and entailed by the adverse events data shown, which contains no reported events. The answer is entailment, as the content or lack of content in the adverse events section supports the claim made in the statement.
PALM
Entailment. The text states that "Total: 0/0". This means that the adverse events section in the primary trial is empty.

Figure 11: Responses from the top-performing LLMs, demonstrating high intelligence and deliberate reasoning.

FALCON 40B
Here, the statement is that "The adverse events section in the primary trial is empty". Since the section provided does contain information (Adverse Events 1 and a total of 0/0), even if no specific adverse events are listed, it cannot logically be considered empty. Therefore, based on reviewing the details shared, the statement that "The adverse events section in the primary trial is empty" would contradict the content presented in the PRIMARY TRIAL ADVERSE EVENTS.
GPT-3.5
Based on the given adverse events section from the primary trial, the statement The adverse events section in the primary trial is empty" entails from the trial information. Here's the justification: In the adverse events section provided for the primary trial, the data is presented as follows: Adverse Events 1: Total: 0/0. The total number of adverse events reported is indicated as 0/0. This suggests that no adverse events were reported in the primary trial. Since there are no adverse events mentioned or reported, it can be inferred that the adverse events section in the primary trial is indeed empty.
LLAMA 2 70B
The Adverse Events section in the primary trial is shown to be empty, with a total of 0/0 adverse events reported. Therefore, the statement is true and entails from the given information.
MIXTRAL 8x7B
Entailment. The adverse events section of the primary trial does not list any adverse events, indicating that no adverse events were recorded during the trial.

Figure 12: Responses from the low-performing LLMs, where all models, except for Falcon 40B, achieved success comparable to the top performers.

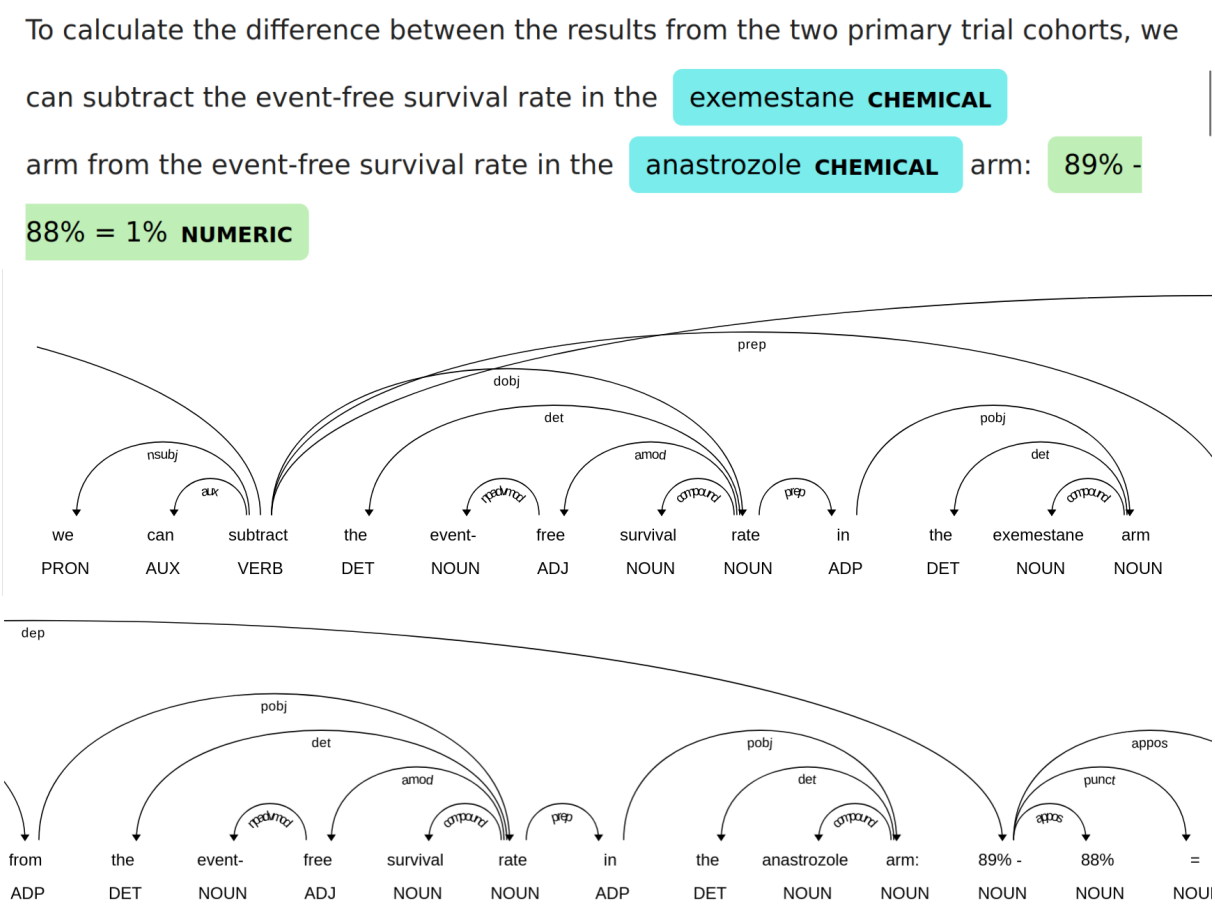


Figure 13: Semantic parse of a successful answer by Gemini. Named entities are highlighted in the above picture, where dependency tree of the sentence is exhibited in the below pictures. In the dependency tree, head token of the numerical expression **89% - 88% = 1%** is **89%** and syntactic head of **89%** is **subtract**, which is the mathematical operation. By following the syntactic parent of the numerical expression, we reach the explanation of the chain of mathematical operations, hence we can deduce that Gemini put down a valid argument and numerical reasoning.

LMEME at SemEval-2024 Task 4: Teacher Student Fusion - Integrating CLIP with LLMs for Enhanced Persuasion Detection

Shiyi Li*, Yike Wang*, Liang Yang†, Shaowu Zhang, Hongfei Lin
Dalian University of Technology
{lishiyieee, yike}@mail.dlut.edu.cn,{liang, zhangsw, hflin}@dlut.edu.cn

Abstract

This paper describes our system used in the SemEval-2024 Task 4 Multilingual Detection of Persuasion Techniques in Memes. Our team proposes a detection system that employs a Teacher Student Fusion framework. Initially, a Large Language Model serves as the teacher, engaging in abductive reasoning on multimodal inputs to generate background knowledge on persuasion techniques, assisting in the training of a smaller downstream model. The student model adopts CLIP as an encoder for text and image features, and we incorporate an attention mechanism for modality alignment. Ultimately, our proposed system achieves a Macro-F1 score of 0.8103, ranking 1st out of 20 on the leaderboard of Subtask 2b in English. In Bulgarian, Macedonian and Arabic, our detection capabilities are ranked 1/15, 3/15 and 14/15.

1 Introduction

Memes are one of the most popular content types in online disinformation activities. They thrive on social media platforms, effortlessly reaching vast audiences. Memes within disinformation activities employ various rhetorical and psychological techniques, such as oversimplification of causation, insults, and defamation, to achieve their impact on users. In this context, meme detection is crucial for identifying and reducing the spread of false information.

The SemEval-2024 Task 4 (Dimitrov et al., 2024) is a multilingual detection task involving persuasion techniques in memes, and we participate in Subtask 2b, the binary classification task, to identify whether it contains a persuasion technique or no technique.In addition to English, the task also includes three test datasets in different languages, which are only released together with the test data in the final phase of the task.The purpose of this design is to evaluate the model's performance in zero-shot scenarios, specifically its capability on languages it has not encountered before.

The key to meme detection lies in uncovering rich correlations within memes between seemingly unrelated text and image components, particularly when there is no apparent connection between the text and image. In cases where the implicit meaning needs deeper exploration and understanding, traditional detection methods often fall short, as they approach meme detection in a straightforward end-to-end manner, overlooking a profound comprehension of meme text and images. Recently, Large Language Models (LLMs) have found success in complex reasoning. They could reveal the underlying implicit meanings beneath the surface of memes, enabling the assessment of whether persuasion techniques are present. Inspired by heuristic teaching (Pintrich and Schunk, 1996), where a teacher with rich experience can impart to students correct thinking and reasoning based on questions and corresponding answers, the students then learn how to deduce their own ways to the correct answers from questions.

To better harness the powerful reasoning capabilities and knowledge reservoir of LLMs, we proposed a Teacher Student Fusion detection system based on the CLIP (Radford et al., 2021) model and LLMs. This system operates in two stages: in the first stage, as the teacher model, the LLM is used to extract prior background knowledge related to persuasion techniques from memes; in the second stage, leveraging this prior knowledge, we fine-tune a smaller student model to detect whether memes contain persuasion techniques.

2 Related Work

2.1 Meme Classification Methods

Meme classification has emerged as a rising multimodal task in recent years. Early multimodal ap-

*These authors contributed equally to this work.
†Corresponding author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 628–633

proaches include models like concatBERT (Kiela et al., 2019), which simply concatenates features from both images and text. Li et al. (2019) introduce the VisualBERT model, touted as the first image-text pretraining model. It utilizes Faster RCNN (Girshick, 2015) for image feature extraction, combines the extracted image features with text embeddings, and then inputs the concatenated features into a single Transformer structure initialized by BERT (Devlin et al., 2018) for classification. Lee et al. (2021) propose the DisMultiHate model, which enhances the classification capability and interpretability of hate memes by introducing entity detection in memes and incorporating statistics on race and gender information as supplementary data. Zia et al. (2021) employ the CLIP encoder to obtain features from both images and text, then simply concatenate these features and pass them to a logistic regression classifier. However, current solutions only capture the superficial signals of different modalities in memes in an end-to-end manner, failing to guide the model in-depth understanding of the complex and diverse relationships between visual and textual elements.

2.2 Large Language Models

Recently, LLMs (Brown et al., 2020) have demonstrated remarkable reasoning capabilities, generating high-quality reasoning steps to augment input prompts to LLMs and improve their few-shot or zero-shot performance (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2022b). Reasoning steps have also been employed for additional fine-tuning to "self-improve" LLMs (Zelikman et al., 2022; Huang et al., 2022). Unfortunately, the large size of LLMs restricts their deployment on detecting memes with diverse modalities. Knowledge distillation has been successfully used to transfer knowledge from larger, more competent teacher models into smaller student models affordable for practical applications (Buciluǎ et al., 2006; Hinton et al., 2015; Beyer et al., 2022). However, existing researches on knowledge distillation from LLMs (Wang et al., 2022a; Ho et al., 2022; Magister et al., 2022) only consider the language modality. To accommodate multimodal features, we conduct abductive reasoning from LLMs, extracting underlying rationales as prompt arguments to assist in meme detection when fine-tuning smaller language models (LMs) for meme detection.

3 System overview

We define a meme detection dataset that potentially contains persuasion techniques as a set of memes where each meme $M = \{I, T, y\}$ is a triplet representing a visual content I that is associated with the textual T, and a ground-truth label $y \in \{propagandistic, non\_propagandistic\}$.

The core idea of our teacher student fusion model is to reason and develop a cognition-level rationale beyond the recognition-level perception (Davis and Marcus, 2015) by constraining the relationships between visual and textual elements in memes. To better utilize multimodal reasoning distilled from LLMs, this task is formulated as a natural language generation paradigm, where our model takes the text T and image I as the input and generates a textual output of the label y to clearly express whether at least one persuasion technique is present in the meme or not. In this paper, we propose to utilize abductive reasoning from LLMs with multimodal inputs to train smaller downstream models. Our overall framework is illustrated in Figure 1, which consists of abductive reasoning from LLMs and model fine-tuning.

3.1 Stage 1: Abductive Reasoning from the Teacher Model

We activate explicit reasoning knowledge in LLMs as the teacher model. Through prompt learning in causal reasoning, the LLM acquires meme-related context and hidden information, to guide student model in detecting persuasion techniques. Given a meme sample M from the training data, we first extract the text caption $\hat{I}$ of the image I to represent the visual content by off-the-shelf captioning model[1]. Specifically, based on the triplet $\left\{\hat{I}, T, y\right\}$, we design a prompt p:

"*Given the textual of a meme: [T], which is embedded in its image: [$\hat{I}$], labeled as [y]. Please provide a streamlined rationale for inferring the meme as [y], incorporating prior background knowledge related to persuasion techniques but without explicitly indicating the label.*"

It prompts the LLMs to generate a rationale R including rich contextual background knowledge, enabling the inference of whether persuasion techniques are present in memes.

[1] https://huggingface.co/nlpconnect/vit-gpt2-image-captioning

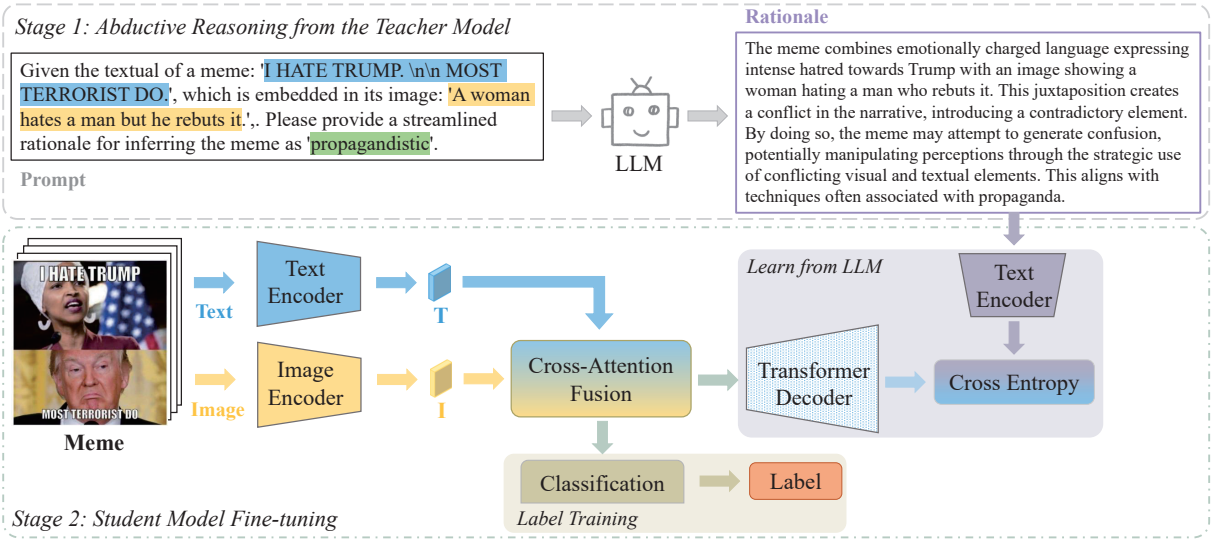


Figure 1: The framework of our method involves two stages. We first conduct abductive reasoning from the teacher model to extract rationales (purple) by the prompt consisting of the meme text (blue), the image caption (yellow), and the label (green). Utilizing the generated rationales, we then train the student model and predict whether memes contain persuasion techniques.

3.2 Stage 2: Student Model Fine-tuning

To facilitate the interaction between meme text and images, we fine-tune a smaller student model for persuasion detection tasks. By using the reasons generated by the teacher model as background knowledge, we aid in uncovering the rich interrelationship between text and vision modalities of memes. For a meme sample M,we first use the encoder of the CLIP model to encode the text T and image I input to obtain the embedding vector H_T and H_I.The advantage of attention mechanism in modality fusion and alignment lies in its ability to dynamically allocate and adjust weights for different modalities, allowing the model to flexibly focus on specific parts of the input. By emphasizing relationships between modalities, the attention mechanism helps improve the effectiveness of modality fusion and enhances the model's ability to capture important information. Therefore, we adopted a cross-attention mechanism for the fusion of textual and visual features.

$$Q_T = W_Q H_T + b_Q \quad (1)$$

$$K_I = W_K\ H_I + b_K \quad (2)$$

$$V_I = W_V\ H_I + b_V \quad (3)$$

$$H_o = Softmax\left(\frac{(K_I)^T Q_T}{\sqrt{d}}\right) V_I \quad (4)$$

Among them, d is the dimension of the feature, softmax is the activation function.

Learn from LLM By inputting the fused features H_o to the transformer decoder for decoding,we obtain the decoded output. Subsequently, we calculate the cross-entropy loss between this output and the given reason, as expressed by the following formula. Minimizing the cross-entropy loss between the meme feature and the generated reason feature facilitates the extraction of prior knowledge from the generated reason. This process helps the model transfer the contextual background information from the generated reason to the meme feature.

$$\mathcal{L}_{llm} = CrossEntropy(decoder(H_o), R) \quad (5)$$

Label Training Through the aforementioned process, our model has acquired the reasoning ability to extract persuasion techniques from LLM. As the objective of the task is to determine whether a meme contains persuasion techniques, label prediction becomes essential. This process shares the same model architecture as the preceding steps, with the introduction of a classifier during the decoding phase for label prediction. During the training process, we optimize the model by minimizing the cross-entropy loss between the predicted labels and the ground truth labels. The loss function is expressed as follows.

$$\mathcal{L}_{label} = CrossEntropy(y^{pre}, y^{true}) \quad (6)$$

Through the above process, for the data samples to be predicted, we can directly predict the labels

Parameter	From LLM	Label Training
Epochs	20	10
Batch size	32	32
Learning rate	5e-4	5e-5
Warmup step	0.1	0.1
Warmup Strategy	Linear	Linear
Image size	224	224

Table 1: Hyper-parameters.

Dataset	Model	Macro-F1
English - Dev	baseline	0.2500
	LMEME(w/o llm)	0.8329
	LMEME	0.8428
English - Test	baseline	0.2500
	LMEME(w/o llm)	0.8043
	LMEME	0.8103

Table 2: Main experimental results of Subtask 2b in English. LMEME is the model proposed in our study. LMEME(w/o llm) represents the ablation results without rationales generated by the teacher model. The baseline refers to the evaluation's benchmark.

of the model without generating corresponding rationales.

4 Experiments

4.1 Dataset and Evaluation

The dataset from the Task 4 of SemEval-2024 contains memes potentially employing persuasion techniques. After training on the English dataset, the model is tested across four languages: English, Bulgarian, North Macedonian and Arabic. As mentioned in the official task description, we employ macro-F1 to evaluate the performance of binary classification Subtask 2b.

In the experiments, we consider the 7b LLaMa2 model (Touvron et al., 2023) as the teacher model. For the task-specific student model, we utilize the CLIP-ViT-B/32 (Radford et al., 2021) architecture as the foundational framework. During the training phase, we evaluate the performance of the model every 100 steps and retain the parameters of the model that performed best on the validation set. The hyperparameters settings adopted are detailed in Table 1. All models are trained on NVIDIA GeForce GTX 3090 GPU.

Dataset	Model	Macro-F1
Bg - Test	baseline	0.1667
	LMEME(w/o llm)	0.6250
	LMEME	0.6710
NM - Test	baseline	0.0909
	LMEME(w/o llm)	0.5536
	LMEME	0.5908
Ar - Test	baseline	0.2271
	LMEME(w/o llm)	0.2933
	LMEME	0.3620

Table 3: Main experimental results in Bulgarian, North Macedonian and Arabic.

4.2 Results and Analysis

Table 2 shows the English detection capabilities of our system in Persuasion Techniques in Memes. A noticeable improvement is observed when comparing it to the scenario without the incorporation of prior knowledge from the LLM, our system in this study demonstrates superior performance. This indicates that leveraging background knowledge obtained from the teacher model can enhance the model's understanding of persuasion techniques to some extent, assisting the model in more accurately detecting memes. Furthermore, the combination of CLIP's encoding capability and the design of cross attention fusion using attention mechanisms enables the system to better align semantic features between text and visuals, facilitating more effective meme persuasion detection.

Table 3 displays our system's zero-shot capability in Bulgarian, North Macedonian, and Arabic. It can be observed that our system exhibits a significant improvement over baseline results in Bulgarian and North Macedonian, ranking 1/15 and 3/15 in the task. In comparison, there is also an improvement in results for Arabic, although not as pronounced as in the first two languages. The potential reason for this lies in the fact that English is an inflectional language with some agglutinative and analytic features. Bulgarian and Macedonian also share some features as inflected languages. However, Arabic introduces agglutinative features onto its inflectional foundation. Since our model has been trained on an English dataset, its effectiveness in detecting agglutinative features might be slightly inferior compared to inflectional languages. Additionally, the ablation results on these

three datasets also indicate the superiority of introducing prior knowledge. This further validates the effectiveness of the teacher student fusion system proposed in this paper.

5 Conclusion

This paper provides a detailed exposition of our approach in addressing Subtask 2b of Semeval2024 Task 4. Our teacher student fusion system initially leverages the Large Language Model as the teacher model to generate background knowledge regarding whether memes contain persuasion techniques. Subsequently, we incorporate this knowledge to fine-tune the student model by minimizing cross-entropy loss, sharing learned parameters with the predictive parameters of the model. Finally, we proceed with predictions using the trained model.

In the future, we will explore alternative encoding methods, better aligned image and textual semantic fusion methods, LLMs of different sizes and types, and different ways of prompting LLMs to generate enhanced background knowledge for meme persuasion detection.

References

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. 2022. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10925–10934.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. 2019. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Roy Ka-Wei Lee, Rui Cao, Ziqing Fan, Jing Jiang, and Wen-Haw Chong. 2021. Disentangling hate in online memes. In *Proceedings of the 29th ACM international conference on multimedia*, pages 5138–5147.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*.

Paul R Pintrich and Dale H Schunk. 1996. Motivation in education: Theory, research, and applications.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Peifeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2022a. Pinto: Faithful language reasoning using prompt-generated rationales. *arXiv preprint arXiv:2211.01562*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Haris Bin Zia, Ignacio Castro, and Gareth Tyson. 2021. Racist or sexist meme? classifying memes beyond hateful. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 215–219.

Innovators at SemEval-2024 Task 10: Revolutionizing Emotion Recognition and Flip Analysis in Code-Mixed Texts

Abhay Shanbhag*, **Suramya Jadhav***, **Shashank Rathi**
Siddhesh Pande, Dipali Kadam
Pune Institute of Computer Technology
{abhayshanbhag0110, 2018suramyajadhav, shashankrathi2}@gmail.com,
siddheshspande@gmail.com, ddkadam@pict.edu

Abstract

In this paper, we introduce our system for all three tracks of the SemEval 2024 EDiReF Shared Task 10, which focuses on Emotion Recognition in Conversation (ERC) and Emotion Flip Reasoning (EFR) within the domain of conversational analysis. Task-Track 1 (ERC) aims to assign an emotion to each utterance in the Hinglish language, a code-mixed language between Hindi and English, from a predefined set of possible emotions. Tracks 2 (EFR) and 3 (EFR) aim to identify the trigger utterance(s) for an emotion flip in a multiparty conversation dialogue in Hinglish and English text, respectively. For Track 1, our study spans both traditional machine learning ensemble techniques, including Decision Trees, SVM, Logistic Regression, and Multinomial NB models, as well as advanced transformer-based models like XLM-Roberta (XLMR), DistilRoberta, and T5 from Hugging Face's transformer library. In the EFR competition, we developed and proposed two innovative algorithms to tackle the challenges presented in Tracks 2 and 3. Specifically, our team, Innovators, developed a standout algorithm that propelled us to secure the 2nd rank in Track 2, achieving an impressive F1 score of 0.79, and the 7th rank in Track 3, with an F1 score of 0.68.

1 Introduction

With advancements in science and technology, the rise of social media has increased remote conversations with different people, resulting in a great deal of linguistic diversity. India is the country with the highest number of users on multiple social media platforms like Facebook, WhatsApp, Instagram, etc. Hinglish remains the most widely used code-mixed language on social media platforms.

A primary challenge associated with code-mixed languages revolves around the misidentification of parts of speech (POS) Atrey et al., 2012. This issue arises when individuals attempt to simultaneously utilize the vocabulary of both languages, leading to the failure of current state-of-the-art machine learning algorithms. Another significant problem identified in code-mixed language is the absence of context within conversations. Unlike traditional emotion detection ML models for pure languages, where a single sentence might suffice to detect emotion, this approach proves inadequate for code-mixed languages like Hinglish. In Hindi-based conversations, context plays a pivotal role in determining emotion Bansal and Lobiyal, 2021.

The data provided by the organizers of SemEval 2024 Task 10 Kumar et al., 2024 for the task comprised conversational episodes, each containing multiple utterances from different speakers. For Track 1 Kumar et al., 2023b, the data included a list of speakers and their utterances, with emotion being the target variable. In contrast, Track 2 Kumar et al., 2022 and Track 3 Kumar et al., 2023a provided utterances and emotions, with triggers as our target variable. Upon examining the training data, we identified an imbalance in the emotion classes, particularly illustrated in Table 1. To address this discrepancy, we applied a range of sampling techniques to effectively rectify the imbalance. Further details about the data are discussed in Section 2.

For Track 1, we employed two approaches: ensemble methods and the transformer approach. In the ensemble methods, we utilized classic ML models such as Random Forest, SVM, Multinomial Naive Bayes, and Logistic Regression, complemented by hyperparameter tuning. For our transformer approach, our main strategy involved creating a pipeline consisting of two main parts: the first deals with converting Hinglish to English, and the second detects emotion from the English output provided by the first. Thus, the pipeline takes Hinglish as input and outputs the corresponding emotions.

For tracks 2 and 3, where we had to detect emotion flips in Hinglish and English conversations, respectively, we developed an algorithm that identifies the last emotion flip of every user. The algorithm takes entire episodes as input and outputs the presence of triggers.

Upon evaluating our approach on the testing set with F1-score as the evaluation metric, we obtained a score of 0.28 for Track 1, 0.79 for Track 2, and 0.68 for Track 3.

The rest of the paper is organized as follows: Section 2 talks discusses the dataset provided by organizers for all three tracks , and Section 3 deals with existing research for several code-mixed tasks focusing on Hinglish text. Further in the paper, we discuss our

*first author, equal contribution

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 634–641

EMOTION	TRAIN	TEST	VALID
Neutral	3,909	656	633
Joy	1,596	349	228
Sadness	819	155	126
Anger	558	142	118
Fear	542	122	88
Contempt	514	82	74
Surprise	441	57	66
Disgust	127	17	21
TOTAL	8,506	1,580	1,354

Table 1: Figure showing distribution and count of emotions for Track 1.

proposed solutions in Section 4. Section 5 gives the experimental setup .Then Section 6 describes the performance of the different approaches along with key findings. Finally in Section 7 we have concluded our discussion.

2 Background

The dataset provided for Track 1 was supplied by the organizers. It consisted of episodes, each containing several sets of utterances in Hinglish. For every utterance, the dataset included the speaker responsible for the utterance, all formatted in JSON. Table 2 offers a glimpse into the Track 1 dataset for one of the episodes.

For Track 2, the data was similar but included an additional column for triggers. A trigger was set to 1 for the last emotion flip of every speaker, while it remained 0 for all other utterances. The primary distinction for Track 3 was the language of the utterances, which was English.

Upon analyzing the dataset, we identified eight emotions: Neutral, Joy, Sadness, Anger, Fear, Contempt, Surprise, and Disgust.

In addition to the organizer's data, we utilized the Hinglish-Top dataset. This dataset features two columns: English (en) and Hinglish (hi-en). We primarily employed this dataset for the Hinglish-to-English conversion component within our pipeline architecture.

3 Related Work

The task of emotion detection and classification has been extensively researched in the context of monolingual data. However, studies focusing on code-mixed text, especially in Indian languages like Hindi mixed with English, are limited due to the scarcity of sufficient data and the absence of a standardized approach for processing code-mixed text.

Foundational research on emotion identification within social media content written in a code-mixed Hindi-English pattern was conducted by Sasidhar et al., 2020. They compiled a dataset of 12,000 code-mixed Hindi-English texts from various sources, annotating them with emotions such as happiness, sadness, and anger. Their study utilized feature vectors generated by a pretrained multilingual model, and the classification models were derived from deep neural networks. Notably, the CNN-BiLSTM approach achieved a classification accuracy of 83.21%, outperforming other models tested in their research.

Wadhawan and Aggarwal, 2021 introduced a deep learning-based technique to recognize emotions in Hindi-English code-mixed tweets. This technique leverages transformer-based models along with bilingual word embeddings produced by Word2Vec and FastText techniques. Their experimentation with CNNs, LSTMs, bi-directional LSTMs, and a variety of deep learning models and transformers, including BERT, RoBERTa, and ALBERT, revealed that the transformer-based BERT model surpassed all others, achieving an accuracy of 71.43% according to their findings.

Bohra et al., 2018 focused on detecting hate speech in social media content that mixes Hindi and English codes, using two distinct classifiers: the Random Forest Classifier and the Support Vector Machines (SVMs). Due to the large feature vectors generated by their study, they employed the chi-square feature selection technique to reduce the size of their feature vector to 1,200. Their findings indicated that SVMs, when utilizing all attributes, outperformed the Random Forest classifier with a maximum accuracy of 71.7% . Additionally, they discovered that Word N-Grams were more effective with the Random Forest Classifier, while Character N-Grams achieved the best results in SVM.

Patil et al., 2023 conducted a comparative analysis of numerous transformer-based language models pre-trained through unsupervised methods, focusing on Hindi and English with mixed codes. Their study included non-code-mixed models such as AlBERT, BERT, and RoBERTa, as well as code-mixed models like HingBERT, HingRoBERTa, HingRoBERTa-Mixed, and mBERT. Models based on HingBERT, specifically trained on authentic code-mixed text, yielded state-of-the-art results on related datasets.

Employing the SentiMix code-mixed dataset, Ghosh et al., 2023 proposed a transformer-based multitask framework for sentiment identification and emotion classification. They enhanced the pre-trained cross-lingual embedding model, XLMR, using task-specific data to improve overall efficiency and leverage transfer learning more effectively.

Singh, 2021 discusses the outcomes of various methods used for sentiment analysis on Hinglish-written social media content, with Twitter serving as a primary example. The data was converted using Fasttext embeddings, count vectorizers, one hot vectorizers, doc2vec, word2vec, and tf-idf vectorizers. Singh employed a range of machine learning techniques, including SVM, CNN, Decision Trees, Random Forests, Naïve Bayes, Logistic Regression, and ensemble voting classifiers, to create the models. The evaluation was based on the F1-score (macro), with the ensemble voting classifier achieving the highest F1-score of 69.07%.

Speaker	Utterances	Emotions
Indu	Wo great hoga! Thanks!	Joy
Monisha	Me abhi tumhare liye new bana deti hun!	Joy
Indu	momma! hath chhodiye dad!	Sad
Monisha	Oh no! Kya hua?	Sad
Indu	Aaj to bhot awful day tha!	Sad

Table 2: Utterances Example from training

		Train	Test	Valid
TRACK 2	No. of episodes	4,893	385	389
	No. of utterances (unique in brackets)	98,777 (10,460)	7,690 (3,650)	7,642 (3,577)
	Avg. utterances per episodes (approx.)	20	20	20
TRACK 3	No. of episodes	4,000	1,002	426
	No. of utterances (unique in brackets)	35,000 (7,831)	8,642 (2,107)	3,522 (924)
	Avg. utterances per episodes (approx.)	9	9	8

Table 3: Track 2 and Track 3 episode-emotion distribution

4 System Description

4.1 Transformer Approach

To translate Hinglish to English and subsequently identify emotions from the translated text, we have developed a two-stage pipeline leveraging the power of transfer learning and pre-trained models from Hugging Face

In the first stage, we utilize the model developed by sayanmandal [1] as our foundational model from Hugging Face. This choice was motivated by its initial proficiency in translating between Hindi and English. To tailor its capabilities more closely to our Hinglish dataset, we applied transfer learning techniques, training it on the Hinglish TOP. dataset[2] by Agarwal et al., 2023.This process resulted in a notable improvement in translation accuracy, as evidenced by achieving a BLEU score of 18.0863%. The model adeptly takes Hinglish as input and outputs the corresponding English text, laying the groundwork for the subsequent emotion analysis.

For the second stage, the English text output from the first model is processed to extract emotional context. We employed the model of j-hartmann [3] from Hugging Face as the baseline for this task. Originally, this model, based on distilRoBERTa, was trained on a diverse array of datasets sourced from Twitter, Reddit, student self-reports, and TV dialogue utterances. However, it did not include 'contempt' among the eight emotion classes specified by the project's guidelines. Therefore, we adapted and further trained this model to recognize the additional emotion class, ensuring a comprehensive analysis of the emotional spectrum in the translated English text.

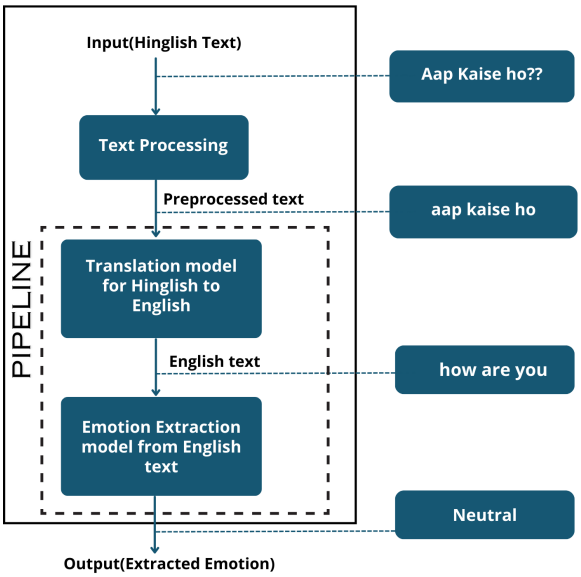


Figure 1: Transformer Architecture along with an example

4.2 XLM-Roberta

XLM-Roberta Conneau et al., 2020 has the ability to process text in Hinglish, a smooth blend of Hindi and En-

[1] sayanmandal/t5-small_6_3-hi_en-to-en
[2] Hinglish TOP dataset
[3] j-hartmann/emotion-english-distilroberta-base

glish, since it is proficient in over 100 languages, including Hindi and English. Its deep linguistic knowledge, reinforced by 2.5 terabytes of training data, enhances its comprehension of Hinglish's emotional nuances. In our work, we trained XLM-Roberta on a particular Hinglish emotion detection dataset using pre-trained weights. It was able to perform better and comprehend Hinglish emotions better as a result.XLMR model helped us to improve the overall performance significantly.

4.3 T5

T5 Raffel et al., 2020 demonstrates an impressive ability to comprehend the subtleties of Hinglish, a language that combines Hindi and English.It served as a good option for translating Hinglish because of its encoder-decoder architecture, which can easily handle code-switching, non-standard syntax, and transliteration. In our work, we fine-tuned the T5 model proposed by $sayanmandal$[2] on Hugging face with hyperparameters given in Table 6 on the external Hinglish TOP Dataset [2], which comprises 3,92,439 translations of Hinglish text into English. As a result, the model outperformed generic models in its ability to comprehend the particular complexities and differences in the dataset.

4.4 Distilroberta

DistilRoBERTa is computationally efficient and perfect for evaluating the frequently enormous amounts of translated text data because it is smaller as compared to RoBERTa and is capable of recording long-range dependencies in text. DistilRoBERTa was pre-trained on two enormous text corpora: BookCorpus and the English Wikipedia making the model more exposed to a wider range of linguistic patterns and improving its understanding of the semantic relationships found in text, both of which help the model identify different emotions. We used the j-hartmann [3] model of Hugging Face in our approach to recognize emotion from translated Hinglish text to English because of its inherent ability to recognize emotions. This helped us navigate any possible emotional nuances that were offered during translation, which strengthened our pipeline approach and increased the accuracy of the detection.

4.5 Random Forest

In Random Forest every tree conducts an independent examination of the data and makes predictions using predetermined feature criteria. A majority vote among all trees determines the final decision, providing resistance against noise and overfitting.Based on certain features like Word frequencies, part-of-speech tags, and sentiment lexicons, the model branches out and divides the input recursively until it reaches leaf nodes, which stand for expected emotions. The layered structure allows you to investigate the characteristics that contribute most to various emotion categories, providing you with a certain level of interpretability. Furthermore, we received higher results from the Random Forest Cutler et al., 2012 trials than from several other methods.

4.6 SVM

Support Vector machines (SVMs) are a useful tool for emotion identification applications because they can quickly scan high-dimensional text input and generate respectable results even with a limited amount of training data. SVMs Evgeniou and Pontil, 1999 excel at determining which feature space hyperplane most effectively separates different emotional classes, capturing the key characteristics that set each emotion apart.SVM proved to have a pretty decent F1 score as compared to other ensemble methods. It is so because of its robust hyperplane-based classification approach. The algorithm then uses statistical techniques to select the optimal line to split the different groups represented in Hearst et al., 1998.

4.7 MNB

Multinomial Naive Bayes (MNB) Kibriya et al., 2005 is a computationally efficient method for handling large datasets with great appropriateness. Using word frequency, it determines the likelihood that a text belongs to each emotion class. The steps in MNB include calculating the likelihood of every word, utilizing the Bayes theorem, and normalizing the probabilities. The final probabilities, which indicate the likelihood that a text belongs to each emotion, are produced by subtracting the estimated probability for each class from the total of the probabilities for all classes.

4.8 Logistic Regression

The linear classification model Logistic Regression Maalouf, 2011 offers a trustworthy and understandable solution for our emotion detection challenge. To forecast the likelihood of each emotion class, it uses a linear combination of input features extracted from the data. The objective variable (or output) in a classification problem, y, can only accept discrete values for a specific set of features (or inputs), X Cox, 1958. Only when a decision threshold is added does logistic regression transform into a classification technique based on the sigmoid function.

4.9 UnderSampling and Oversampling

In our experimental endeavors, we explored both oversampling and undersampling techniques Mohammed et al., 2020 to address class imbalances within our training dataset. The necessity for such interventions became evident as the 'neutral' class dominated the dataset—outnumbering instances of emotions like 'sad' and 'anger' by nearly double, and 'disgust' by an astounding factor of thirty. This disproportion threatened to skew the learning process, potentially biasing the model towards the overrepresented classes.

To mitigate this, we employed oversampling strategies, particularly focusing on the minority classes. By replicating instances from these underrepresented categories, we aimed to achieve a more equitable distribution across all emotions. This technique not only

Algorithm 1

Require: A dictionary of episode data with each entry containing a speaker, utterances, and the emotion associated with that speaker.

Ensure: A list indicating the trigger points, where each trigger point is set to 1.0 in case of a flip trigger and 0.0 elsewhere.

Initialization:
Initialize **context**: A dictionary to store each speaker's emotions and their indices like {emotion : indices}.
Initialize **lastFlipForEverySpeaker**: An empty list to store the indices of the last emotion change for each speaker.
Build Context:
for each speaker in the episode data **do**
    if the speaker is not in context **then**
        initialize their context
    end if
    append a dictionary {emotion: index} to the context for the current speaker
end for
Identify Last Emotion Changes:
for each speaker in the context **do**
    Initialize **lastFlip** to 0 and **lastEmo** to 'null'.
    for each emotion index in the speaker's context **do**
        Extract emotion and index from the context.
        if lastEmo is not equal to emotion **then**
            Set lastFlip to index.
            Set lastEmo to emotion.
        end if
    end for
    Append lastFlip - 1 to lastFlipForEverySpeaker.
end for
Initialize Trigger List:
Initialize **trig** as a list of 0.0s with a length equal to the number of speakers.

Mark Trigger Points:
for each speaker index **do**
    if the speaker's index is in lastFlipForEverySpeaker **then**
        set the corresponding element in trig to 1.0.
    end if
end for
return trig as the list of trigger points.

Algorithm 2

Require: A dictionary episodes with keys 'speakers' and 'labels'.

Ensure: A list of triggers for each episode, where each trigger list has a 1.0 for the second last conversation and 0.0 for the rest.

Algorithm:
Determine the number of speakers in the episode.
Initialize an empty list named **trig** to store trigger flags.
for each speaker in the episodes['speakers'] list **do**
    if the speaker is not the second-to-last one **then**
        append "0.0" to the trig list, indicating a non-trigger condition.
    else
        append "1.0" to the trig list, indicating a trigger condition.
    end if
end for
return a tuple consisting of the trig list from the episodes dictionary.

prevented the majority class from monopolizing the learning dynamics but also ensured that the model received ample exposure to each emotion. As a result, the capability of our model to accurately recognize emotions that were previously underrepresented saw significant improvement. Conversely, undersampling was also considered a method to harmonize the dataset. This approach involves reducing the instances of the majority class to match the numbers of the minority classes, thereby leveling the playing field. However, while undersampling can effectively reduce bias towards overrepresented classes, it also entails the risk of losing valuable information by discarding data.

Overall, we concluded that oversampling helped in the training process by giving each emotion class equal weight, and unlike undersampling, there was no loss of data.

4.10 Metrics Used F1 Score

For evaluating our model, we used the F1 score as our metric, which is given as the harmonic mean of precision and recall.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (1)$$

Here, precision is the number of samples correctly predicted out of the number of samples predicted in that category. Recall is the number of samples predicted correctly out of the number of samples present for that class.

5 Experimental Setup

5.1 Data Preprocessing

Data preprocessing steps like lowercasing,stopword removal,punctuation removal and stemming were per-

Figure 2: Data Preprocessing Overview

formed before feeding text data into ensemble learning methods as shown in Figure 2. Emotion classification improves model performance by reducing noise, normalizing text, and enhancing feature extraction because of data prerocessing.

5.2 Vectorisation for ensemble methods

Word2Vec translates words into numerical vectors that represent their relationships to other words as well as their meanings. These vectors are valuable because they convey semantic understanding rather than merely indicating word presence. CBOW Mikolov et al., 2013, which predicts words based on context, and Skip-gram are two significant Word2Vec architectures. In our experiment, Word2Vec was utilized to convert processed text into vectors, which were then inputted into ensemble learning models for emotion prediction.

6 Result

	Track1	Track2	Track3
Our F1 Score	0.28	0.79	0.68
Our Rank	27	2	7
Max F1 Score	0.78	0.79	0.79

Table 4: Leaderboard Results

6.1 Key Findings

Our models demonstrated enhanced efficiency when the input data was augmented using minority oversampling. The input data exhibited a significant class imbalance, leading the model to predominantly recognize the dominant class. To address this issue, we employed both undersampling and oversampling techniques. Oversampling notably improved model performance, as indicated in Table 5, because it enabled the model to learn about the minority classes more effectively. We adjusted all feature sizes to match that of the dominant class size (in this case, 'neutral').

As illustrated in Figure 3 and Figure 4, our Algorithm 1 approach for Tracks 2 and 3 showed a considerable number of false positives, or negative samples incorrectly predicted as positive, amounting to 959. This figure was substantially higher than that observed in Algorithm 2, which was only 68. The count of false negatives in Algorithm 1 was comparable to that in Algorithm 2 (68 and 99, respectively), though Algorithm 1

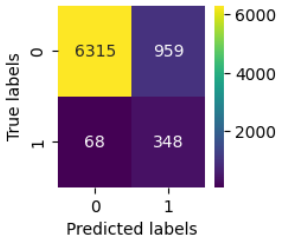


Figure 3: Confusion Matrix for Algorithm 1 on test data of track 2

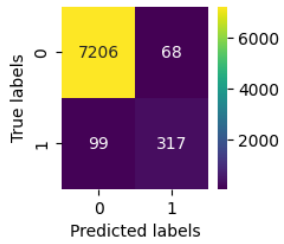


Figure 4: Confusion Matrix for Algorithm 2 on test data of track 2

was slightly more effective than Algorithm 2 in reducing false negatives.

In summary, instances of being erroneously classified as 0s were marginally higher in Algorithm 2, whereas instances of 0s being wrongly classified as 1s were significantly higher in the case of Algorithm 1.

7 Conclusion

In our participation in SemEval 2024 Task 10, we embraced two approaches: first, ensemble methods, and next, a transformer pipeline for our experiments in Track 1. Our analysis revealed a compelling insight: even marginal enhancements in translation accuracy can lead to substantial improvements in emotion classification outcomes. This underscores not only the importance of the sentence itself but also the critical role of contextual understanding in accurately leveraging this foundational insight. We developed and proposed two distinct algorithms designed to adeptly navigate the challenges of emotion flip recognition in Tracks 2 and 3.

Furthermore, our experiments highlighted the effectiveness of oversampling as a strategy to counteract the dataset's imbalance—a challenge characterized by a striking 30:1 ratio between dominant and minority classes. This technique emerged as a performance enhancer, enabling our models to achieve a more balanced understanding and representation of all emotional classes. Through these methodical and strategic efforts, we contributed valuable insights to the field and also demonstrated our algorithms' potential to transform emotion recognition practices.

APPROACH	F1 SCORE
DT	0.2495
DT (Undersampled)	0.2255
DT (Oversampled)	0.2578
SVM	0.2297
SVM (Undersampled)	0.2602
SVM (Oversampled)	**0.2830**
Multinomial Naive Bayes	0.1945
MultinomialNB (Undersampled)	0.2209
MultinomialNB (Oversampled)	0.2623
Logistic Regression - Softmax	0.2242
Logistic Regression - Softmax (Undersampled)	0.2584
Logistic Regression - Softmax (Oversampled)	0.2809
Logistic Regression - OvR	0.2242
Logistic Regression - OvR (Undersampled)	0.2584
Logistic Regression - OvR (Oversampled)	**0.2809**
Random Forest Classifier	0.2418
XLMR Approach	0.2626
Pipeline Approach	**0.2688**

Table 5: F1 Scores for Different Approaches used in Track 1

Parameter	Value
learning_rate	2e-05
train_batch_size	32
eval_batch_size	64
seed	42
gradient_accumulation_steps	2
weight decay	0.01
optimizer (Adam with betas)	(0.9, 0.999)
epsilon	1e-08
lr_scheduler_type	linear
num_train_epochs	100
mixed_precision_training	Native AMP

Table 6: Hyperparameters for Fine Tuning

References

Anmol Agarwal, Jigar Gupta, Rahul Goel, Shyam Upadhyay, Pankaj Joshi, and Rengarajan Aravamudhan. 2023. CST5: Data augmentation for code-switched semantic parsing. In *Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants!*, pages 1–10, Prague, Czech Republic. Association for Computational Linguistics.

Shree Atrey, T. Prasad, and G. Krishna. 2012. Issues in parsing and pos tagging of hybrid language. pages 20–24.

Mani Bansal and D. Lobiyal. 2021. Context-based machine translation of english-hindi using ce-encoder. *Journal of Computer Science*, 17:825–843.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of hindi-english code-mixed social media text for hate speech detection. In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pages 36–41.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2):215–232.

Adele Cutler, D Richard Cutler, and John R Stevens. 2012. Random forests. *Ensemble machine learning: Methods and applications*, pages 157–175.

Theodoros Evgeniou and Massimiliano Pontil. 1999. Support vector machines: Theory and applications. In *Advanced Course on Artificial Intelligence*, pages 249–257. Springer.

Soumitra Ghosh, Amit Priyankar, Asif Ekbal, and Pushpak Bhattacharyya. 2023. Multitasking of sentiment detection and emotion recognition in code-mixed hinglish data. *Knowledge-Based Systems*, 260:110182.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2005. Multinomial naive bayes for text categorization revisited. In *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia,*

December 4-6, 2004. Proceedings 17, pages 488–499. Springer.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023a. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, pages 1–10.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023b. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Maher Maalouf. 2011. Logistic regression in data analysis: an overview. *International Journal of Data Analysis Techniques and Strategies*, 3(3):281–299.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013.

Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. 2020. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)*, pages 243–248. IEEE.

Aryan Patil, Varad Patwardhan, Abhishek Phaltankar, Gauri Takawane, and Raviraj Joshi. 2023. Comparative study of pre-trained bert models for code-mixed hindi-english data. In *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, pages 1–7. IEEE.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

T Tulasi Sasidhar, Premjith B, and Soman K P. 2020. Emotion detection in hinglish(hindi+english) code-mixed social media text. *Procedia Computer Science*, 171:1346–1352. Third International Conference on Computing and Network Communications (CoCoNet'19).

Gaurav Singh. 2021. Sentiment analysis of code-mixed social media text (hinglish). *ArXiv*, abs/2102.12149.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in Hindi-English code-mixed data: A transformer based approach. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 195–202, Online. Association for Computational Linguistics.

DUTIR938 at SemEval-2024 Task 4: Semi-Supervised Learning and Model Ensemble for Persuasion Techniques Detection in Memes

Erchen Yu[1], Junlong Wang[2], Xuening Qiao[1], Jiewei Qi[1], Zhaoqing Li[1], Hongfei Lin[1], Linlin Zong[2], Bo Xu[1*]

[1] School of Computer Science and Technology, Dalian University of Technology, China

[2] School of Software, Dalian University of Technology, China

{yuerchen0809, jlwang, qiao, 1329027682, lizhaoqing}@mail.dlut.edu.cn

{hflin, llzong, xubo}@dlut.edu.cn

Abstract

The development of social platforms has facilitated the proliferation of disinformation, with memes becoming one of the most popular types of propaganda for disseminating disinformation on the internet. Effectively detecting the persuasion techniques hidden within memes is helpful in understanding user-generated content and further promoting the detection of disinformation on the internet. This paper demonstrates the approach proposed by Team DUTIR938 in Subtask 2b of SemEval-2024 Task 4. We propose a dual-channel model based on semi-supervised learning and model ensemble. We utilize CLIP to extract image features, and employ various pretrained language models under task-adaptive pretraining for text feature extraction. To enhance the detection and generalization capabilities of the model, we implement sample data augmentation using semi-supervised pseudo-labeling methods, introduce adversarial training strategies, and design a two-stage global model ensemble strategy. Our proposed method surpasses the provided baseline method, with Macro/Micro F1 values of 0.80910/0.83667 in the English leaderboard. Our submission ranks 3rd/19 in terms of Macro F1 and 1st/19 in terms of Micro F1.

1 Introduction

Social networks play a significant role in our society. The development of social platforms has facilitated the dissemination of information, but it has also fueled the proliferation of disinformation (Da San Martino et al., 2020; Dimitrov et al., 2021). The dissemination mechanism of disinformation involves the use of propaganda techniques. "Propaganda" is defined as a dissemination pattern referring to stakeholders influencing public opinion to support specific agendas and ideas by adopting persuasion techniques, such as disseminating one-sided, biased, or even fake news. Research on detecting propaganda techniques contributes to combating network disinformation (Da San Martino et al., 2021).

Among all types of content on social networks, memes play a significant role. Memes typically exist in the form of images, possibly with overlapping text, and convey information in the form of jokes, irony, etc. In the current era of social media, they spread rapidly and can influence many people without awareness. Memes are one of the most popular content types in online disinformation propaganda activities and serve as a powerful medium for promoting ideological and cognitive persuasion techniques (Moody-Ramirez and Church, 2019). Therefore, research on automatically detecting persuasion techniques hidden in memes is of significant importance, which contributes to understanding user-generated content and further aids in detecting network disinformation.

Subtask 2b of Semeval-2024 Task 4 aims to promote research on computational methods to detect persuasion techniques in memes (Dimitrov et al., 2024), which is modeled as a binary classification problem. Due to the complexity and subjectivity inherent in persuasive language, single multimodal model may struggle to capture all relevant features effectively. To address this issue, we propose a dual-channel model based on semi-supervised learning and model ensemble in this paper. Within the image channel, we use CLIP (Radford et al., 2021) to extract image features from memes. Concurrently, in the text channel, we employ diverse pretrained language models and conduct task-adaptive pretraining utilizing certain corpora provided by the task. We execute feature extraction from the pretrained language model, employing a variety of methodologies to capture sentence features, followed by a concatenation and fusion process of the extracted features, subsequently fed into the classification layer. We implement a semi-supervised pseudo-labeling ap-

*Corresponding Author

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 642–648

proach, wherein pseudo-labels are assigned to the test set, thereby augmenting the training data to achieve data augmentation. Furthermore, to bolster each model's robustness, We employ Fast Gradient Method(FGM) as our adversarial training strategy, introducing perturbations to the embedding layer of the model. Lastly, We design a two-stage soft-voting ensemble strategy to amalgamate the predictions of multiple models, augmenting the model's generalization capacity and performance.

We applied our proposed method to the English dataset. Our proposed method ranked 3rd/19 in terms of Macro F1 and 1st/19 in terms of Micro F1 in the English leaderboard for Subtask 2b.

2 Related Work

Transformer (Vaswani et al., 2017) is a deep learning model based on the self-attention mechanism, which is known for its ability to effectively capture long-range dependencies in sequential data. Based on Transformer, pretrained language models such as BERT (Kenton and Toutanova, 2019) have been proposed, which capture the contextual information of each token in the text through self-attention. Subsequent pretrained language models have mainly been modified on pretraining tasks, such as RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2020), and so on. These text models improve upon BERT by enhancing semantic feature representation in text feature extraction.

Existing image feature extraction methods mostly rely on multiple convolutional neural networks (Li et al., 2021), such as VGG (Simonyan and Zisserman, 2015) and ResNet (He et al., 2016). Vision Transformer (ViT) (Dosovitskiy et al., 2020) is an image processing model based on the traditional Transformer, dividing the input image into image patches and using multi-head self-attention mechanisms to capture global relationships between images. ViT represents a significant advancement of Transformer in the field of computer vision, bringing new ideas and methods to image processing tasks. CLIP (Radford et al., 2021) is a multimodal model which is trained through contrastive learning on 400 million pairs of images and text. CLIP achieves high-performance cross-modal semantic feature extraction for images and text.

3 System Overview

In this section, we will introduce the overall structure of our proposed system. Our system is divided into two stages. (1) Training of our Text-Image Multi-modal Classification Model. During the training process, we introduce several training strategies, including task-adaptive pretraining, pseudo-labeling and adversarial training. (2) Model Ensemble. We use k-fold cross-validation for model training and design a two-stage soft-voting strategy to globally integrate the models obtained in the first stage.

3.1 Model Architecture

The architecture of our model is shown in Figure 1. Our proposed text-image multi-modal classification model can be divided into two modules: feature extraction module and cross-modal fusion module. In the feature extraction module, we employ a parallel architecture to perform feature extraction separately for image and text channels using pretrained models.

For image inputs, we utilize the pretrained CLIP model. Before inputting images into the CLIP image encoder, they are resized and normalized, resulting in one-dimensional features of dimensionality 512. For text inputs, we experiment with various pretrained language models and their variants, including BERT, RoBERTa, DeBERTa, XLM (Conneau et al., 2020), DistilBERT (Sanh et al., 2019), etc. Based on their performance on the validation and dev sets, we ultimately select two general domain models and two models pretrained using political domain related corpora as text encoder models: CLIP text encoder, Deberta-v3-large (He et al., 2022), politicalBiasBERT (Baly et al., 2020), and xlm-twitter-politics-sentiment (Antypas et al., 2023). We select the latter two models because the vast majority of persuasion techniques are reflected in political domain.

The output of the CLIP text encoder is a one-dimensional vector of 512 dimensions, while for three BERT-like models above, the encoder's output is a two-dimensional vector that needs to be converted into a one-dimensional vector through pooling operation. Common pooling operations include cls, pooler, last layer average and first-last layer average.The optimal pooling method is selected for each BERT-like model as the model's pooling strategy. After the pooling layer, the text features are ultimately obtained as one-dimensional features of 512 dimensions.

The cross-modal fusion module aims to integrate features from two modalities. We concatenate the output image features with the text features and em-

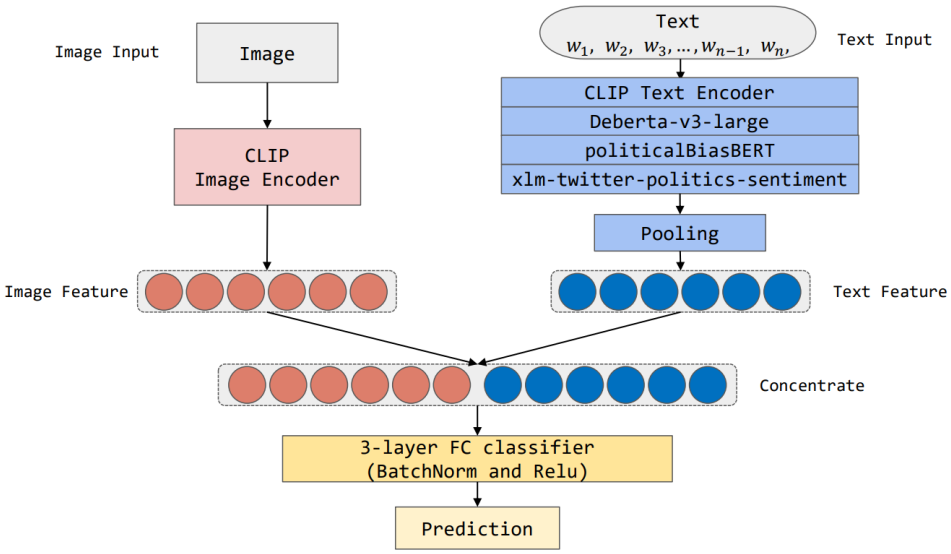


Figure 1: The architecture of our Text-Image Multi-modal Classification Model.

ploy a three-layer fully connected network as the classifier to map the final representation obtained from the fusion layer to a scalar, which is then bounded between 0 and 1 through a sigmoid function. The formula of cross-modal fusion module is defined as:

$$\hat{y}_c = \text{sigmoid}\left(W\left[h^I, h^T\right] + b\right)$$

Where h^I is image feature and h^T is text feature. Finally, the model would be fit with the binary cross-entropy loss function.

3.2 Training strategies

To further improve the performance of our model, we adopt three training strategies:

Task-adaptive Pretraining(TAPT): Certain research has proved that further pretraining models on the unlabeled task data itself or task related data, called TAPT, can improve the performance of model in downstream tasks (Gururangan et al., 2020). In this task, we adopt TAPT for three BERT-like text models. The purpose of this task is to detect persuasion techniques in memes. Therefore, we perform TAPT on text models using the provided dataset. Specifically, we collect and utilize all texts from task1, task2a, and task2b for pretraining our models with masked language model(MLM). By conducting MLM pretraining, these models can not only better fit the distribution in the task, but also learn rich knowledge and semantic information, thereby performing better on the task.

Pseudo-labeling: Pseudo-labeling is a semi-supervised method aimed at predicting labels for unlabeled data using a model trained on labeled data and adding the labeled data to the training set to achieve data augmentation. Considering the diversity of samples in the test sets, we adopt a semi-supervised pseudo-labeling method for data augmentation. We train four models on the original training set and then ensemble the four trained models using soft-voting for inference on unlabeled samples, i.e., test set samples. Subsequently, we consider samples with output probabilities greater than or equal to 0.8. Finally, we obtain 393 pseudo-labeled test set samples, which are re-incorporated into the training set for training.

Adversarial Training: Adversarial training is a common method to improve the robustness of neural networks. By adding perturbations in the embedding layer, we can obtain more stable embedding representations and more universal models, improving the performance of models on unseen data. In this task, we introduce FGM (Miyato et al., 2016) to enhance the model's robustness. The adversarial perturbation δ on s is defined as:

$$\delta = \epsilon \cdot g/\|g\|_2 \quad where \quad g = \nabla_s L(s, y)$$

where ϵ is a constant that controls the degree of perturbation suppression. The idea of FGM is to increase the perturbation direction along the gradient, where increasing along the gradient means maximizing the loss.

3.3 Ensemble Learning

To integrate the learning abilities of each model and improve the generalization ability of the final sys-

Dataset	Negative	Positive
Train	400	800
Validation	50	100
Dev	100	200
Test(Pseudo-labeled)	101	292
All	651	1392

Table 1: The label distribution of the dataset in task2b. Negative means non-propagandistic and positive means propagandistic.

Setting	Value
Epochs	20
Max Sequence Length	128
Batch Size	16
Optimizer	Adam
Learning Rate	5e-5
Dropout	0.5
Weight Decay	0.001

Table 2: Hyper-parameter settings of the experiment.

tem, we design a two-stage soft-voting strategy to ensemble the four models saved from k-fold cross-validation. In this task, due to the small size of the dataset, we use k-fold cross-validation to train models, ensuring that all data participate in training and validation to effectively avoid over-fitting. In the first stage, we average output probability values generated by the four models in each fold of the k-fold cross-validation, resulting in the probability values for each fold. In the second stage, we aggregate the probability values from each fold by averaging, yielding the final output. We determine the optimal binary classification threshold by testing on the validation set for various thresholds.

4 Experimental setup

4.1 Data description and Evaluation

The dataset is provided by Subtask 2b of Semeval-2024 Task 4. In the original dataset partition, there are 1200 samples in the training set, 150 samples in the validation set, 300 samples in the dev set, and 600 samples in the test set. After pseudo-labeling, 101 samples in the test set are labeled as non-propagandistic and 292 samples are labeled as propagandistic. We aggregate the training set, validation set, dev set, and the test set augmented with pseudo-labels into a new dataset for k-fold cross-validation. The label distribution of this dataset is shown in Table 1.

The official evaluation metrics for this task are Macro F1 and Micro F1, with a focus primarily on the performance of Macro F1 in our experiments. Both Macro F1 and Micro F1 performances are presented in the final test set ranking.

4.2 Implementation

During validation, we conduct 8-fold cross-validation on the dataset and consistently use the average measure from the first fold as the new validation set to evaluate the performance of our model. We preserve the model parameters to achieve optimal performance. During the testing phase, we train each model separately and predict the test set through the two-stage soft-voting strategy we purposed, resulting in the final prediction by averaging the probabilities from all trained models in 8 folds. After comparing the overall performance under different thresholds, we set the final threshold as 0.5, which yields the optimal average performance on the validation set under 8-fold cross-validation.

We implement our model using the transformer package[1]. We select the following four models as text encoder models: CLIP text encoder, Deberta-v3-large, xlm-twitter-politics-sentiment and politicalBiasBERT. And we select CLIP ViT-L/14@336px as the image encoder model. We sequentially combine each text encoder with the image encoder, and four final models are sequentially as follows: CLIP_{img}+$\text{CLIP}_{\text{text}}$, CLIP_{img}+$\text{DeBERTa}_{\text{text}}$, CLIP_{img}+XLM_{text} and CLIP_{img}+$\text{BERT}_{\text{text}}$. As for the pooling method, we test various options and selected the optimal one for each model based on performance on the validation set. Specifically, CLIP_{img}+$\text{DeBERTa}_{\text{text}}$ and CLIP_{img}+$\text{BERT}_{\text{text}}$ performed optimally with first-last layer average pooling, while CLIP_{img}+XLM_{text} performs optimally with pooler. We employ early stopping to retain the model parameters that exhibited the best performance on the validation set. Details of the hyper-parameter settings are provided in Table 2. We used a weighted binary cross-entropy loss using the class distribution. By default, We set ϵ to 1.0 in FGM. All experiments are conducted on an RTX 4090 with 24GB of memory.

5 Results

The overview statistics of four different models on the new validation set are shown in Table 3. Among all base models, CLIP_{img}+$\text{CLIP}_{\text{text}}$

[1]https://huggingface.co/

Setting	$CLIP_{img}$+$CLIP_{text}$	$CLIP_{img}$+$DeBERTa_{text}$	$CLIP_{img}$+XLM_{text}	$CLIP_{img}$+$BERT_{text}$
Base	0.8524	0.8375	0.8505	0.8385
+FGM	**0.8534**	0.8555	0.8435	0.8455
+TAPT	/	**0.8574**	0.8465	0.8505
+FGM+TAPT	/	0.8515	**0.8515**	**0.8586**

Table 3: Macro F1 for four models on the new validation set. "Base" indicates no training strategy added, "+FGM" indicates the addition of FGM, "+TAPT" indicates the addition of TAPT.

performs best, and $CLIP_{img}$+XLM_{text} likewise exhibits impressive performance. Additionally, $CLIP_{img}$+$BERT_{text}$+FGM+TAPT attains the highest Macro F1 score of 0.8586. The politicalBiasBERT model with FGM and TAPT strategies showcased its robust reasoning capability in detecting persuasion techniques in memes.

Experimental results highlight the effectiveness of TAPT and FGM strategies. Introducing FGM to the $CLIP_{img}$+$CLIP_{text}$, $CLIP_{img}$+$DeBERTa_{text}$ and $CLIP_{img}$+$BERT_{text}$ lead to significant performance enhancements. Both the $CLIP_{img}$+$DeBERTa_{text}$ and $CLIP_{img}$+$BERT_{text}$ models demonstrate performance improvements after introducing TAPT, and $CLIP_{img}$+$DeBERTa_{text}$with TAPT achieves the second-best overall performance. When both TAPT and FGM are employed simultaneously, all three models with BERT-like text encoder demonstrate substantial performance improvements compared with base model, while $CLIP_{img}$+XLM_{text} and $CLIP_{img}$+$BERT_{text}$ achieve their respective optimal performance levels, showcasing the effectiveness of adding FGM and TAPT to base model.

Regarding model ensemble, our analysis, illustrated in Table 4, demonstrates a significant enhancement in performance with the adoption of our two-stage global integration approach. Moreover, the scalability and robustness of our integration method are validated by the observed performance scalability relative to the number of integrated models. Results shows that all model ensemble performed best, which demonstrates the effectiveness of our model ensemble strategy.

We employed all-model ensemble as the final model of our system and submitted the result file of the test set predicted by the final model. The official rankings are shown in Table 5. We ranked 3rd in terms of Macro F1 and 1st in terms of Micro F1. Results show that our system demonstrates outstanding performance in detecting persuasion techniques in memes, and the integration of TAPT, FGM and model ensemble techniques further enhances the detection capability of our system.

Method	Macro F1	Micro F1
$CLIP_{img}$+XLM_{text}	0.8515	0.8711
Two-Model Ensemble	0.8515	0.8711
Three-Model Ensemble	0.8596	0.8789
All-Model Ensemble	**0.8654**	**0.8789**

Table 4: Results for model ensemble on the new validation set. Two-model ensemble refers to integrating $CLIP_{img}$+XLM_{text} and $CLIP_{img}$+$BERT_{text}$. Three-model ensemble adds $CLIP_{img}$+$DeBERTa_{text}$. For all-model ensemble, all four models are integrated for ensemble.

Rank	Team	Macro F1	Micro F1
1	LMEME	0.81030	0.82500
2	SuteAlbastre	0.80964	0.83500
3	**DUTIR938**	**0.80910**	**0.83667**
4	BCAmirs	0.80337	0.82500
5	Snarci	0.79860	0.82667

Table 5: Results of top 5 teams for subtask2b English leaderboard on the test set.

6 Conclusion

The paper presents our system designed for Subtask 2b of Semeval-2024 Task 4. We propose a dual-channel model based on semi-supervised learning and model ensemble. Our framework leverages multiple pretrained models served as feature extractors for images and texts. We integrate a semi-supervised pseudo-labeling approach for data augmentation, and introduce TAPT and FGM adversarial training to significantly enhance the model's performance and robustness. Finally, to enhance the generalization capability of our system, we design a two-stage soft-voting model ensemble strategy. Our system achieves excellent performance in detecting persuasion techniques in memes, and we ranked 3rd in terms of Macro F1 and 1st in terms of Micro F1 in the test set for Subtask 2b. Our future research will be directed towards exploring the cross-modal fusion mechanisms within the model.

7 Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work was supported by grant from the Natural Science Foundation of China(N0.62006034), the Ministry of Education Humanities and Social Science Project (No.22YJC740110), the Fundamental Research Funds for the Central Universities (No.DUT23YG136).

References

Dimosthenis Antypas, Alun Preece, and Jose Camacho-Collados. 2023. Negativity spreads faster: A large-scale multilingual twitter analysis on the role of sentiment in political communication. *Online Social Networks and Media*, 33:100242.

Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. 2020. We can detect your bias: Predicting the political ideology of news articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4982–4991.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. Semeval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2021. A survey on computational propaganda detection. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4826–4832.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Semeval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

Mia Moody-Ramirez and Andrew B Church. 2019. Analysis of facebook meme groups used during the 2016 us presidential election. *Social Media+ Society*, 5(1):2056305118808799.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

K Simonyan and A Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

ISDS-NLP at SemEval-2024 Task 10: Transformer based neural networks for emotion recognition in conversations

Claudiu Creangă[2,3], **Liviu P. Dinu**[1,3]
[1] Faculty of Mathematics and Computer Science
[2] Interdisciplinary School of Doctoral Studies, [3] HLT Research Center
University of Bucharest, Romania
claudiu.creanga@s.unibuc.ro, ldinu@fmi.unibuc.ro

Abstract

This paper outlines the approach of the ISDS-NLP team in the SemEval 2024 Task 10: Emotion Discovery and Reasoning its Flip in Conversation (EDiReF). For Subtask 1 we obtained a weighted F1 score of 0.43 and placed 12 in the leaderboard. We investigate two distinct approaches: Masked Language Modeling (MLM) and Causal Language Modeling (CLM). For MLM, we employ pre-trained BERT-like models in a multilingual setting, fine-tuning them with a classifier to predict emotions. Experiments with varying input lengths, classifier architectures, and fine-tuning strategies demonstrate the effectiveness of this approach. Additionally, we utilize Mistral 7B Instruct V0.2, a state-of-the-art model, applying zero-shot and few-shot prompting techniques. Our findings indicate that while Mistral shows promise, MLMs currently outperform them in sentence-level emotion classification.

1 Introduction

Task 10 from SemEval 2024 competition (Kumar et al., 2024) addresses the complex challenge of identifying the emotions within dialogues (English and Hindi). This task comprises two primary objectives: firstly, assigning an emotion label to each utterance within a dialogue, and secondly, discerning the trigger utterance or utterances responsible for an emotion-flip within the dialogue (Kumar et al., 2022). Emotions play a crucial role in human interaction and one can understand more from a text if one knows the underlying sentiment of the writer. In contexts where disagreements may arise, such as customer service platforms, virtual assistant chats or forums, identifying trigger utterances for emotion flips can help mediate conflicts and prevent escalation. A chatbot dealing with an angry customer would benefit from knowing how to speak in order to generate empathetic responses. If it knows that the chatbot's current sentence can trigger an emotion flip from neutral to anger, the chatbot should refine it, or if the emotion flip is from anger to joy, the chatbot should be more confident in such a response in the future.

Both types of models we tried for Subtask 1 were based on transformers. The first one used BERT-like models and we achieved the best accuracy with them, while the second one is a state of the art causal model (Mistral, (Jiang et al., 2023)) that was tested in zero-shot and few-shot settings with poorer results.

Although in the first task our system worked well, placing 12th in the leaderboard, the other 2 tasks were much harder and we placed 14th on the second subtask. We believed that with a better strategy to prevent overfitting (like under or oversampling), our system would have improved. Our code is open source and available to use on GitHub.

2 Background

The competition had 3 subtasks explained in Figure 1 and we participated in all of them with the best results on subtask 1 where we placed 12th with an F1 score of 0.43.

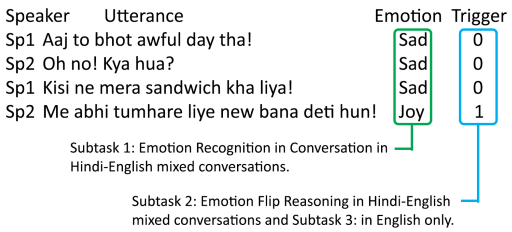


Figure 1: Three sub-tasks explained

2.1 Dataset

The dataset contains English and Hindi code-mixed conversations for Subtask 1 and 2 and English only conversations for Subtask 3 (Table 1). The dataset is quite small, except for the training dataset for

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 649–654

Table 1: Datasets sizes used in this competition by tasks.

	Subtask 1	Subtask 2	Subtask 3
Train	8506	98777	35000
Dev	1354	7462	3522
Test	1580	7690	8642

Subtask 2 and 3. If we were to combined them for Subtask 1, our F1 score would reach 0.97, but it wasn't allowed. This fact shows that with more data our model would do really well. The dataset is based on MELD, a known emotion recognition dataset, which was then augmented with triggers for the emotion-flip task.

There were 8 distinct emotions to predict: neutral, anger, surprise, fear, joy, sadness, disgust, and contempt. By far the most predominant emotion is neutral, followed by joy and anger (Figure 2). If we look at Subtask 2, most often the emotion flips are from neutral to joy or anger (Figure 3).

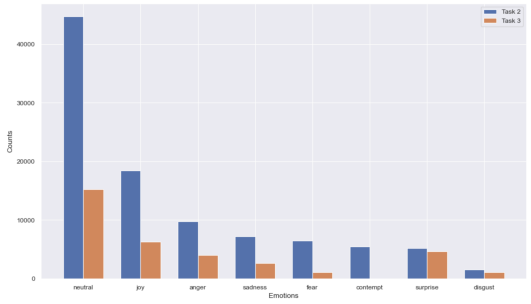


Figure 2: Emotion Distribution Comparison between Task 2 and Task 3

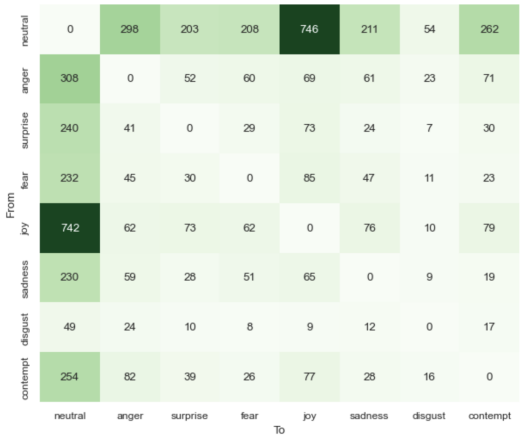


Figure 3: Task 2: Emotion-flip counts

2.2 Previous Work

Since the release of the first small datasets for emotion recognition in 1992 (Ekman, 1992), the field has evolved substantially, marked by significant contributions from big companies in the form of extensive datasets (Demszky et al., 2020). In the beginning, lexicon based methods were used in which there was a manually curated dictionary which associates words with specific emotions. The algorithm was simply picking the most expressed emotion according to the dictionary. This method had severe limitations because it was ignoring context, sentence structure and negations which can flip a sentiment. Today, state of the art models are based on transformer architecture and use either Masked Language Modelling (BERT based models (Devlin et al., 2018)) or Causal Modelling (GPT (Brown et al., 2020)) which can capture dependencies and nuances missed by word-level approaches.

While traditional emotion recognition tasks are well-established, research on emotion flip recognition is still in its early stages because it is a new task within the field of emotion analysis. Research (Kumar et al., 2021) has found that a transformer based classifier with 6 encoder layer (EFR-TX) works well, obtaining an F1 score of 40 when trained on MELD-FR dataset and tested on IEMOCAP-FR dataset.

3 System overview

We tried two approaches, both of them based on transformer architecture: Masked Language Modelling and Casual Modelling. We chose these two architectures because of their recent successes in NLP.

3.1 Masked Language Modelling

We used pre-trained BERT-like models in a multilingual setting so that it can tokenise Hindi sentences. These pre-trained models will give us the features from sentences and then we pass them through a classifier which will do the prediction for each task Figure 4.

Figure 4: Model architecture

3.1.1 Input

Analysis of dialogue sentences reveals a predominantly short length, with a sharp decline in frequency after 30 tokens (see Figure 5). To optimize performance, various maximum sequence lengths were tested, with 55 tokens yielding the best results (Figure 6). Data preprocessing, (such as lemmatization, removing punctuation or stopwords) didn't help the model learn better so we kept the input as is. Probably this is because punctuation and stopwords contain useful information that the models is able to learn.

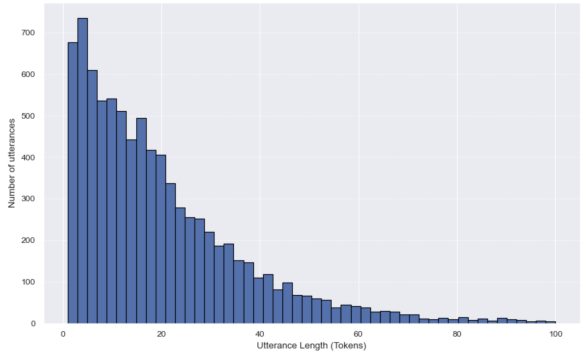


Figure 5: Distribution of utterances lengths.

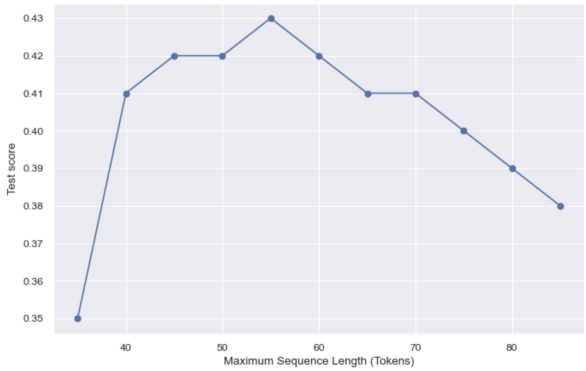


Figure 6: Best model score with different maximum utterances lengths.

3.1.2 Output

Selecting the optimal hidden state layer is crucial for leveraging the pre-trained model's results. Our experiments demonstrated that using the final layer's output yielded the strongest performance, with accuracy declining in earlier layers. For MLM-type models, the [CLS] token encodes the features, which is what we pass to our classification layer.

Among various classifiers tested (Table 2), fully connected layers excelled, likely due to their ability to model complex, non-linear relationships. The top-performing model employed a fully connected layer with 0.5 dropout and a Softmax activation function.

Table 2: Test scores of different classifiers.

Classifier	Extra features	Score
Fully Connected	Dropout(0.5)	0.43
Fully Connected	Dropout(0.7)	0.42
Fully Connected	Dropout(0.2)	0.40
Fully Connected	-	0.40
RandomForest	-	0.23
LogisticRegression	-	0.21
KNeighbors	-	0.20

3.1.3 Fine-tuning

The large pre-trained language models we employed offer a robust foundation for understanding language in general. Through fine-tuning, we adapt them to the nuances of our emotion recognition task. Inspired by the strategy presented in (Sun et al., 2020), we initially train only the classifier with a larger learning rate (5e-5) and a warm-up period of 10,000 steps over 'k' epochs (we tried a range of 'k' from 1 to 10). Subsequently, we fine-tune both the classifier and the transformer's final layer using a smaller learning rate (2e-5). Our goal in freezing the transformer weights at first, and then training them with a reduced learning rate, is to minimize the risk of overfitting.

3.2 Causal Modelling

Given the success of generative models we also tried Mistral 7B Instruct V0.2 which is believed to be state of the art in its category of models (Jiang et al., 2023). These type of LLMs have had success in a large number of NLP tasks, but seem to still lag Masked Language Models in sentence classification.

3.2.1 Prompting

In Causal Modelling, how you prompt the model significantly influences its performance. We tested different prompting strategies in both zero-shot and few-shot settings:

- **Zero-Shot Learning:** Here, we provide the model with a single example and ask it to predict the emotion without any additional references. For zero-shot learning the best prompting technique was: "[INS] Given the following sentence: `{sentence}`. ### Predict which emotion is expressed. Chose one

of the following options: neutral, anger, surprise, fear, joy, sadness, disgust, and contempt. Answer in one word only. Answer: [\INS]"

- **Few-Shot Learning:** In this setting, we give the model several examples – one for each emotion – along with their corresponding labels. This leverages the model's in-context learning ability, potentially boosting its performance for unseen samples. For few-shot learning the best prompting technique was: "[INS] This is an example of a sad sentence: {sentence} {repeat for every emotion}. ### Predict the emotion of the following sentence: sentence. Chose one of the following options: neutral, anger, surprise, fear, joy, sadness, disgust, and contempt. Answer in one word only. Answer: [\INS]"

4 Experimental setup

4.1 Data Split Strategy

We employed a classic data split approach:

- **Initial Development**: We combined the training and development sets and shuffled the data. Subsequently, we used 70% for training, 10% for validation, and the remaining 20% as a held-out test set.
- **Competition Test Set Release**: Upon the competition's test set release, we directly evaluated our models using the platform. To maximize training data, we trained on the combined training set with a 20% validation split.
- **Final Model**: Once we selected our best model, we re-trained it on the entire dataset without validation. This re-training didn't yield significant improvements

4.2 Subtask 1

We'll focus on Subtask 1, where we achieved strong results. The key hyperparameters used:

- **Batch Size**: A batch size of 64 provided the best balance. Smaller sizes hurt performance, while larger sizes exceeded our memory constraints.
- **Fine-Tuning**: We trained for 4 epochs with frozen model weights, followed by 3 epochs with only the last layer unfrozen (as detailed in section 3.1.3).
- **Classifier**: Our classifier used 128 neurons, 0.5 dropout, and a softmax activation.
- **Optimization**: We used cross-entropy loss, the AdamW optimizer, and experimented with different learning rates (see section 3.1.3).
- **Evaluation**: We measured performance using the MulticlassF1Score with 8 classes and 'macro' averaging.

5 Results

Our top-performing model (Table 3) was a fine-tuned FacebookAI/xlm-roberta-large (Conneau et al., 2019). This highlights the superiority of fine-tuned Masked Language Models (MLMs) over Mistral for sentence classification tasks. The results suggest that smaller Causal models remain less effective than fine-tuned MLMs in this domain. We also see that few-shot Mistral is worse than zero-shot, probably because too much data in the prompt confuses the model.

Model	Train	Validation	Test
xlm-roberta	0.74	0.57	**0.43**
mdeberta-v3	0.74	0.56	0.42
bert-multi	0.66	0.48	0.35
Mistral zero-shot	-	-	0.32
Mistral few-shot	-	-	0.31
distilbert-multi	0.6	0.47	0.29

Table 3: Results for Subtask 1 - Masked Language Models and Causal Models (Mistral).

In terms of number of epochs, our best model was overfitting when finetuned for too many epochs (Table 4) and we finally trained for 4 + 3 epochs.

Frozen	Fine-tunning	Training	Validation	Test
3	2	0.67	0.53	0.4
3	3	0.71	0.55	0.42
4	3	0.74	0.57	**0.43**
4	4	0.78	0.60	0.42
4	5	0.85	0.45	0.38
5	3	0.71	0.56	0.42

Table 4: Finding the optimal number of epochs to avoid overfitting. First column contains epochs when training only the classifier. Second columns contains epochs when training the classifier and the last transformer layer.

5.1 Error analysis

Our confusion matrix (Figure 7) reveals that the model overpredicts the 'neutral' emotion, likely due to its prevalence in the training data. This created a bias, leading the model to misclassify instances of other emotions as 'neutral'. While we attempted to mitigate this with class weights in the loss function, it proves insufficient. In the future, we should explore more robust techniques like oversampling or undersampling to address the class imbalance.

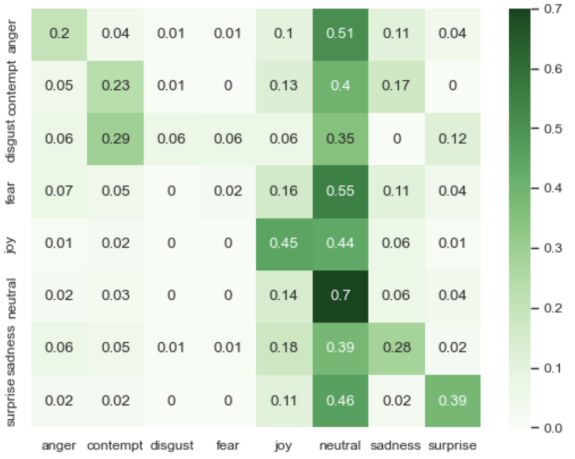


Figure 7: Confusion matrix. On y-axis true labels, on x-axis predicted labels. Values are normalised.

As seen in the emotion accuracy chart (Figure 8), the model performs best on the dominant 'neutral' class, along with well-represented emotions like 'joy' and 'sadness'. Conversely, the model struggles to predict the 'disgust' emotion, which aligns with its under-representation in the training data. This suggests a direct correlation between dataset frequency and model proficiency for each emotion.

6 Conclusion

Overall, our system achieved encouraging results in Subtask 1, despite exhibiting some overfitting for dominant labels. While performance on the emotion-flip detection tasks (Subtasks 2 and 3) highlights areas for improvement, we still placed in the first half of the leaderboard. Looking ahead, we plan to investigate hybrid transformer-LSTM architectures for a more nuanced understanding of emotion-flip triggers. Additionally, enriching the data by incorporating a broader conversational context through multi-turn analysis could enhance our model's capabilities. Not least, even though we tried Mistral, there are newer causal models like

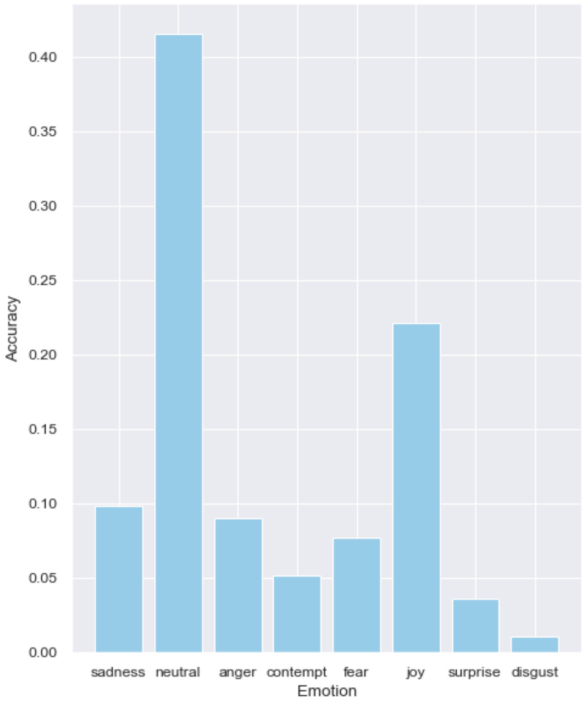


Figure 8: Accuracy by emotion. Accuracy directly correlates with the frequency of each emotion in the training set.

Mixtral (Jiang et al., 2024) and Solar (Kim et al., 2023) which could perform better at this type of task.

Acknowledgements

This work was partially supported by a grant on Machine Reading Comprehension from Accenture Labs and by the POCIDIF project in Action 1.2. "Romanian Hub for Artificial Intelligence".

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6(3–4):169–200.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2023. SOLAR 10.7B: Scaling large language models with simple yet effective depth up-scaling.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2021. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification?

UMUTeam at SemEval-2024 Task 4: Multimodal Identification of Persuasive Techniques in Memes through Large Language Models

Ronghao Pan[1], **José Antonio García-Díaz**[1], **Rafael Valencia-García**[1]
[1] Facultad de Informática, Universidad de Murcia, Campus de Espinardo, 30100 Murcia, Spain
{ronghao.pan, joseantonio.garcia8, valencia}@um.es

Abstract

In this manuscript we describe the UMUTeam's participation in SemEval-2024 Task 4, a shared task to identify different persuasion techniques in memes. The task is divided into three subtasks. One is a multimodal subtask of identifying whether a meme contains persuasion or not. The others are hierarchical multi-label classifications that consider textual content alone or a multimodal setting of text and visual content. This is a multilingual task, and we participated in all three subtasks but we focus only on the English dataset. Our approach is based on a fine-tuning approach with the pre-trained RoBERTa-large model. In addition, for multimodal cases with both textual and visual content, we used the LMM called LlaVa to extract image descriptions and combine them with the meme text. Our system performed well in three subtasks, achieving the tenth best result with an Hierarchical F1 of 64.774%, the fourth best in Subtask 2a with an Hierarchical F1 of 69.003%, and the eighth best in Subtask 2b with a Macro F1 of 78.660%.

1 Introduction

The rise of social media has facilitated the rapid spread of information. However, its unconstrained nature has also led to the spread of information whose accuracy is difficult to verify. As a result, misinformation and disinformation have become serious problems in everyday life. For example, during the COVID-19 pandemic, social media enabled healthcare professionals to quickly communicate professional information to the public; however, studies also revealed the spread of inaccurate health-related information (Ferrara et al., 2020).

A special case of spreading misinformation is the use of memes. Memes consist of images overlaid with text created by Internet users and have become one of the primary forms of content in online disinformation campaigns. Designed specifically to actively spread inaccurate information, "disinformation memes" are particularly effective on social media platforms, where they can quickly reach large audiences (Qu et al., 2022). Using various rhetorical and psychological techniques such as causal oversimplification, name-calling, and smear tactics, memes play a pivotal role in influencing users' perceptions and beliefs.

To address this phenomenon, the Multilingual Detection of Persuasion Techniques in Memes shared task has been organized at SemEval-2024 (Dimitrov et al., 2024). The goal of this task is to develop models for detecting persuasion techniques in the textual content of a meme, as well as in a multimodal setting where both textual and visual content are analyzed together. The task is divided into three main subtasks:

- **Subtask 1**. This is a unimodal hierarchical multi-label classification. The goal is to identify which of the 20 persuasion techniques are present using only textual features.
- **Subtask 2a**. This is a multimodal hierarchical multi-label classification. The goal is to identify which of the 22 persuasion techniques are present using textual and visual multimodal features.
- **Subtask 2b**. This is a multimodal binary persuasion identification task, where the goal is to determine whether a meme contains a persuasion technique or not.

To solve the English challenge, we propose an approach based on fine-tuning Transformer models for binary and hierarchical multi-label classification problems of persuasion techniques using textual and visual content. During training for subtasks 2a and 2b, we used a Large Multimodal Model (LMM) called LlaVa (Liu et al., 2023) to extract textual and visual features from the memes. We then refined the monolingual model, as RoBERTa-large (Liu et al., 2019), to identify persuasion techniques and their type.

In multimodal classification problems, our experiments showed that including the textual description of the meme obtained by an LMM improves the overall performance. In our experiments, this strategy achieved better results and required fewer resources than merging the image and text embeddings into the same vector space.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 655–666

The rest of this paper is organized as follows. Section 2 provides a summary of important details about the task setup. Section 3 provides an overview of our system for two subtasks. Section 4 presents the specific details of our systems. Section 5 discusses the results of the experiments, and finally the conclusions are presented in section 7.

2 Background

Recently, there has been a significant increase in the use of memes on social media as a means of spreading misinformation. Memes consist of a combination of text and images that together have a meaning that is very difficult to automatically verify. In addition, the image and text of the meme in isolation may convey a benign meaning, but their combination may be derogatory, or vice versa. Fake news and hate speech purveyors use memes as a tool to spread misinformation and hateful content. They may spread hate to create unrest among the people, and such hateful content may target communities or individuals based on religion, ethnicity, race, national origin, affiliation, sexual orientation, gender, sex, disability, and disease (Hamza et al., 2023).

Many studies have focused on identifying memes that contain negative content or misinformation. For example, the authors of (Hamza et al., 2023) published a dataset of religiously hateful memes and evaluated it fine-tuning VisualBERT, which was pre-trained on the Conceptual Caption (CC) dataset for the top-down classification task. Visual features were extracted using ResNeXT-152 Aggregated Residual Transformations based Masked Regions with Convolutional Neural Networks (R-CNN) and BERT without textual encoding for the early fusion model. Regarding multimodal approaches, there have been tasks previously organized in the same area of interest. MAMI (Multimedia Automatic Misogyny Identification) at SemEval-2022 (Fersini et al., 2022), which explored the detection of misogynistic memes on the web using available text and images; and DravidianLangTech at EACL-2021 (Suryawanshi and Chakravarthi, 2021), which explored the detection of offensive language and classification of troll memes.

The novelty of this shared task is the focus on disinformation propaganda through memes. Propaganda uses psychological and rhetorical techniques to achieve its goal. These techniques include the use of logical fallacies and appeals to the audience's emotions. Logical fallacies are often difficult to detect because the argument seems correct and objective at first glance. However, careful analysis reveals that the conclusion cannot be deduced from the premise without the misuse of logical rules. Therefore, memes are a perfect medium for spreading disinformation because they consist of an image superimposed on text, and the image can be deceptive, reinforcing or complementing one or more persuasive techniques in the text or image. Thus, the goal of this task is to identify the existence and type of persuasion techniques through memes with different subtasks. The persuasion techniques can be viewed on the official task page.[1]. It is worth noting that a similar propaganda technique was used in Dipromats 2023 (IberLEF) (Moral et al., 2023).

The dataset used for this task is the one provided by the organizers. It consists of a set of texts and images labeled with their corresponding persuasion techniques and a binary annotation for Subtask 2b. The data set provided by the organizers is divided into train, dev, and validation. Note that we do not actually need two datasets for validation (dev and validation), since the dev set was used for the development phase. Therefore, we have combined the train and dev sets into a single training set. The training dataset contains 8000 examples for subtasks 1 and 2a and 1499 examples for subtask 2b. Figure 1 shows the distribution of the training set for subtasks 1a and 2a and Figure 2 for subtask 2b.

3 System overview

Figure 3 shows the architecture of our system for the three subtasks. We can see that for Subtask 1, only the text of the memes is used, which is a multi-label classification problem of different persuasion techniques. To address subtask 1, we have fine-tuned RoBERTa-large (Liu et al., 2019). For Subtasks 2a and 2b, we have used a similar approach as in Subtask 1, but including textual and visual features. We rely on LlaVa (Liu et al., 2023) to extract the image description and then concatenate this information with the textual content of the memes, as shown in Figure 4. LlaVa is an end-to-end multimodal Large Language Model (LLM) that incorporates a vision encoder for general pur-

[1] https://propaganda.math.unipd.it/semeval2024task4/

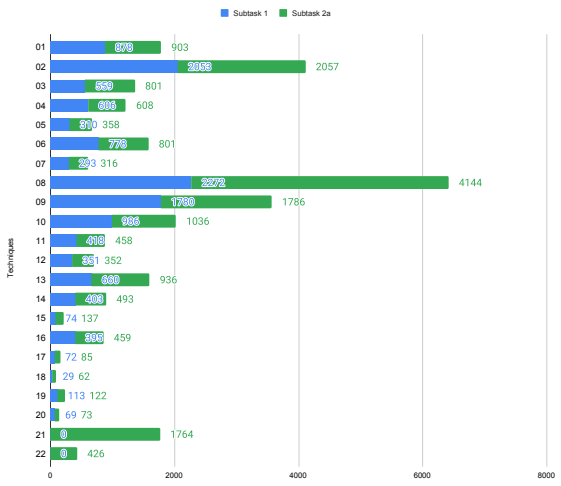


Figure 1: Distribution of training for Subtasks 1 and 2a. The techniques are: (01) Black-and-white Fallacy / Dictatorship; (02) Loaded Language; (03) Glittering generalities (Virtue); (04) Thought-terminating cliché; (05) Whataboutism, (06) Slogans, (07) Causal Oversimplification; (08) Smears; (09) Name calling/Labeling; (10) Appeal to authority; (11) Exaggeration/Minimisation; (12) Repetition; (13) Flag-waving; (14) Appeal to fear/prejudice; (15) Reductio ad hitlerum; (16) Doubt; (17) Misrepresentation of Someone's Position (Straw Man); (18) Obfuscation, Intentional vagueness, Confusion; (19) Bandwagon; (2) Presenting Irrelevant Data (Red Herring); (21) Transfer; (22) Appeal to (Strong) Emotions.

pose visual and language understanding. LlaVa has demonstrated impressive multimodal conversational capabilities, sometimes exhibiting behavior similar to the multimodal GPT-4 on unseen images/instructions, and achieving a relative score of 85.1% compared to GPT-4 on a synthetic multimodal instruction-following dataset (Liu et al., 2023). It is worth noting that the output model as a binary classification problem and in subtask 2b as a multi-class hierarchical classification problem like subtask 2a.

4 Experimental setup

In this work, we used only the dataset provided by the organizers and we did not rely on external data except for the use of LLM and LMM models that were pre-trained with general purpose data.

Before fine-tuning, we performed a preprocessing step to remove line breaks, hashtags, symbols, references, and hyperlinks. Next, for all the subtasks, we performed the fine-tuning process using an epoch-based evaluation strategy with the Hug-

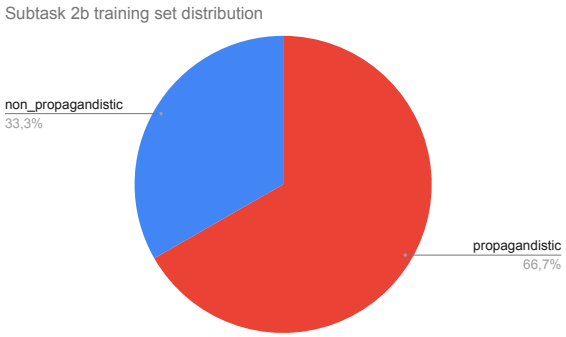


Figure 2: Distribution of training for Subtask 2b.

gingface Trainer library[2]. This involves training the pre-trained model for a certain number of epochs and performing an evaluation with the evaluation set after each epoch. Once all epochs have been completed, the model with the best macro F1 score in the evaluation set is selected. In this way, overfitting or underfitting resulting in low variance and high bias can be avoided.

We used the same hyperparameters for fine-tuning in all the subtasks: (1) a batch size of 8 for both training and validation, (2) 10 epochs, (3) a learning rate of 2e-5, (4) and a weight decay of 0.01. During training, we used macro-F1 as a reference. For the evaluation of subtasks 1 and 2a, the organizers used hierarchical-F1 as the primary evaluation metric, and for subtask 2b, macro-F1. It should be noted that in order to ensure the reproducibility of the experiment, we modified the LlaVa generation configuration by setting the value of `do_sample` to False.

Hierarchical precision, recall and F1 (H-P, H-R, and H-F1) are metrics used in hierarchical classification problems where classes are organized in a hierarchical structure (Kiritchenko et al., 2006). H-F1 considers both precision and recall of the prediction for each class in the hierarchy, taking into account the relationship between parent and child classes in the hierarchy.

The binary task (subtask 2b) is evaluated using the macro F1 score, which is an evaluation metric used in classification problems to measure the precision and recall of a model in predicting multiple classes. It assigns equal weight to each class, meaning that all classes have the same impact on the final metric, regardless of their size or distribution in the data.

[2] https://huggingface.co/docs/transformers/main_classes/trainer

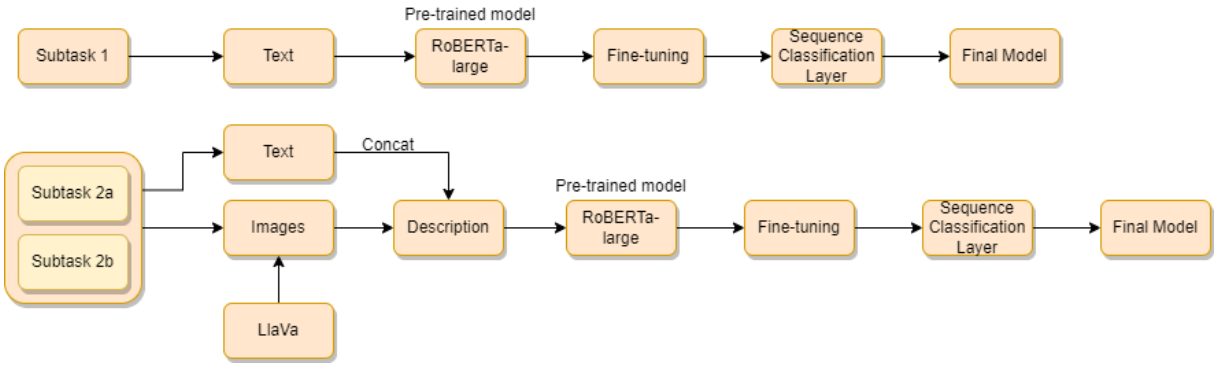


Figure 3: System architecture approach

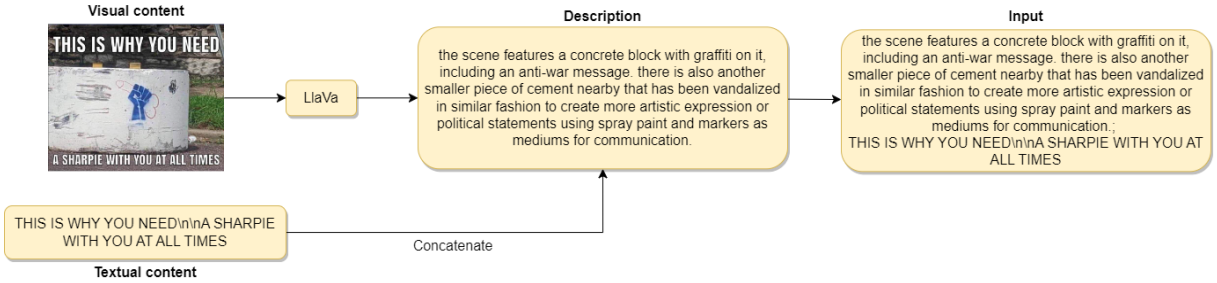


Figure 4: Example of multimodal input for Subtasks 2a and 2b.

5 Results

Table 1 shows the official ranking table for Subtask 1. We can see that we have ranked tenth position with a H-F1 score of 0.64774 and a H-P of 70.817%. We outperformed the baseline by almost 50% in terms of H-F1 and are 10.473% away from the best result, which is 75.247%, achieved by the '914isthebest' team.

Table 1: Official results for the Subtask 1.

Team	Rank	H-F1	H-P	H-R
914isthebest	1	75.247	68.419	83.590
BCAmirs	2	69.857	66.786	73.223
Otterly Obsessed With Semantics	3	69.738	64.801	75.490
TUMnlp	4	67.384	63.781	71.419
GreyBox	5	66.998	65.248	68.844
		...		
UMUTeam	10	**64.774**	**70.817**	**59.681**
		...		
Baseline	-	36.865	47.711	30.036

For Subtask 2a, we achieved a H-F1 of 69.003% and a H-P of 76.763%, which is the fourth-best result according to the official ranking table (see Table 2). With our approach, we outperformed the baseline by 24.297% and are only 5.589% away from first place, which achieved an H-F1 of 74.592%.

Table 2: Official results for the Subtask 2a.

Team	Rank	H-F1	H-P	H-R
Hierarchy Everywhere	1	74.592	86.682	65.461
NLPNCHU	2	70.677	78.164	64.498
BCAmirs	3	70.497	78.374	64.059
UMUTeam	4	**69.003**	**76.763**	**62.669**
		...		
Baseline	-	44.706	68.778	33.116

For subtask 2b, which is a binary classification problem to identify the presence of persuasion techniques in memes, we obtained a macro-F1 score of 78.660%, which puts us in eighth place according to the official ranking table (see Table 3). Furthermore, we can see that our system has improved by up to 53.66% compared to the baseline and is only 2.37% behind the first place (LMEME with a M-F1 of 81.030%).

Based on the results obtained, it's clear that combining image descriptions with textual content im-

Table 3: Official results for the Subtask 2b.

Team	Rank	M-F1	m-F1
LMEME	1	81.030	82.500
SuteAlbastre	2	80.964	83.500
DUTIR938	3	80.910	83.667
BCAmirs	4	80.337	82.500
Snarci	5	79.860	82.667
	...		
UMUTeam	8	**78.660**	**80.667**
	...		
Baseline	-	25.000	33.333

proves overall performance in a multimodal setting. This approach does not impose any restrictions on embedding images and text together in the same vector space when fine-tuning or training persuasion classification techniques. Rather, we merge the text from the meme with its description and use this combined dataset as input for fine-tuning the pre-trained transformer-based model.

6 Error analysis

As far as we know, the organizers did not provide the gold labels of the test set to the participants. Therefore, we did a bug analysis based on the results of the development set.

Table 4 shows the results and the ranking we got with our development set approach.

In subtask 1 our approach is based on a fine-tuned model of RoBERTa-large. We obtained an H-F1 of 62.201 and an M-F1 of 35.514. From the confusion matrices (see Figure 5), we can see that the model didn't correctly predict any instances of the classes `Misrepresentation of Someone's Position (Straw Man)`, `Obfuscation, Intentional vagueness, Confusion`, `Presenting Irrelevant Data (Red Herring)`, and `Reductio ad Hitlerum`, indicating a possible class imbalance or lack of representative features for these classes. In addition, the F1 score of the `Whataboutism` and `Causal Oversimplification` class is relatively low compared to other classes, suggesting that the model has difficulty correctly identifying instances of this class, possibly due to ambiguous or overlapping features with other classes.

Subtask 2a is a hierarchical multi-label classification problem, but unlike Subtask 1, it uses a multimodal dataset, i.e. it uses both textual and visual multimodal features to identify 22 persuasion techniques. In this case, our model achieved an H-F1 of 67.902 and an M-F1 of 36.841, which is an improvement over the unimodal approach (Subtask 1). However, similar to the model in Subtask 1 (see Figure 6), it failed to predict any instances of the classes `Misrepresentation of Someone's Position (Straw Man)`, `Obfuscation, Intentional Vagueness, Confusion`, and `Presenting Irrelevant Data (Red Herring)`, and it obtained a lower F1 score in `Casual Oversimplification` and `Appeal to (Strong) Emotions`, except for the class `Reductio ad Hitlerum`, for which it correctly predicted 2 instances. This could be due to insufficient training data or ineffective feature representation for these classes.

Regarding Subtask 2b, a multimodal binary classification problem, our approach achieved an M-F1 of 76.836, and in Figure 7 we can see that the model misclassified 40% of the examples as non-propagandistic and 9% as propagandistic.

7 Conclusion

In this paper, we describe our participation in a SemEval task focused on identifying persuasive techniques in memes using a multimodal approach. For all three subtasks, we used the fine-tuning approach with the RoBERTa-large model for text features and LlaVa to extract image descriptions and combine them with the meme text. Our system achieved the tenth best result with an H-F1 of 64.774%, the fourth best in Subtask 2a with an H-F1 of 69.003%, and the eighth best in Subtask 2b with a macro-F1 of 78.660%.

As further work, we will evaluate the relationship between the persuasion techniques used in the different domains evaluated by our team. In this sense, we propose to re-annotate the Spanish SatiCorpus 2021 (García-Díaz and Valencia-García, 2022) and the PoliticES 2022 dataset (García-Díaz et al., 2022), which are focus on figurative language and politics respectively, with the 22 persuasion techniques and evaluate the reliability of using binary and hierarchical multi-label classification approaches. Another area where persuasion techniques may be present is in the identification of misogyny (García-Díaz et al., 2023).

Table 4: Results for dev split.

-	Rank	H-F1	H-P	H-R	M-F1	m-F1
Subtask 1	14	62.201	71.111	55.276	35.514	52.439
Subtask 2a	4	67.902	75.151	61.929	36.841	57.124
Subtask 2b	11	-	-	-	76.836	80.667

Acknowledgments

This work is part of the research projects LaTe4PoliticES (PID2022-138099OB-I00) funded by MICIU/AEI/10.13039/50110001103 and the European Regional Development Fund (ERDF)-a way to make Europe and LT-SWM (TED2021-131167B-I00) funded by MICIU/AEI/10.13039/50110001103 and by the European Union NextGenerationEU/PRTR. In addition, this work was funded by the Spanish Government, the Spanish Ministry of Economy and Digital Transformation through the Digital Transformation through the "Recovery, Transformation and Resilience Plan" and also funded by the European Union NextGenerationEU/PRTR through the research project 2021/C005/00149877. Mr. Ronghao Pan is supported by the "Programa Investigo" grant, funded by the Region of Murcia, the Spanish Ministry of Labour and Social Economy and the European Union - NextGenerationEU under the "Plan de Recuperación, Transformación y Resiliencia (PRTR)".

References

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Emilio Ferrara, Stefano Cresci, and Luca Luceri. 2020. Misinformation, manipulation, and abuse on social media in the era of covid-19. *Journal of Computational Social Science*, 3:271–277.

Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. 2022. SemEval-2022 task 5: Multimedia automatic misogyny identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 533–549, Seattle, United States. Association for Computational Linguistics.

José Antonio García-Díaz, Salud M Jiménez Zafra, María Teresa Martín Valdivia, Francisco García-Sánchez, Luis Alfonso Ureña López, and Rafael Valencia García. 2022. Overview of politices 2022: Spanish author profiling for political ideology. *Procesamiento del Lenguje Natural*.

José Antonio García-Díaz, Salud María Jiménez-Zafra, Miguel Angel García-Cumbreras, and Rafael Valencia-García. 2023. Evaluating feature combination strategies for hate-speech detection in spanish using linguistic features and transformers. *Complex & Intelligent Systems*, 9(3):2893–2914.

José Antonio García-Díaz and Rafael Valencia-García. 2022. Compilation and evaluation of the spanish saticorpus 2021 for satire identification using linguistic features and transformers. *Complex & Intelligent Systems*, 8(2):1723–1736.

Ameer Hamza, Abdul Rehman Javed, Farkhund Iqbal, Amanullah Yasin, Gautam Srivastava, Dawid Połap, Thippa Reddy Gadekallu, and Zunera Jalil. 2023. Multimodal religiously hateful social media memes classification based on textual and image data. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*, pages 395–406. Springer.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Pablo Moral, Guillermo Marco, Julio Gonzalo, Jorge Carrillo-de Albornoz, and Iván Gonzalo-Verdugo. 2023. Overview of dipromats 2023: automatic detection and characterization of propaganda techniques in messages from diplomats and authorities of world powers. *Procesamiento del lenguaje natural*, 71:397–407.

Jingnong Qu, Liunian Harold Li, Jieyu Zhao, Sunipa Dev, and Kai-Wei Chang. 2022. Disinfomeme: A

multimodal dataset for detecting meme intentionally spreading out disinformation. *arXiv preprint arXiv:2205.12617*.

Shardul Suryawanshi and Bharathi Raja Chakravarthi. 2021. Findings of the shared task on troll meme classification in tamil. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 126–132.

A Confusion matrices for the error analysis with the test set

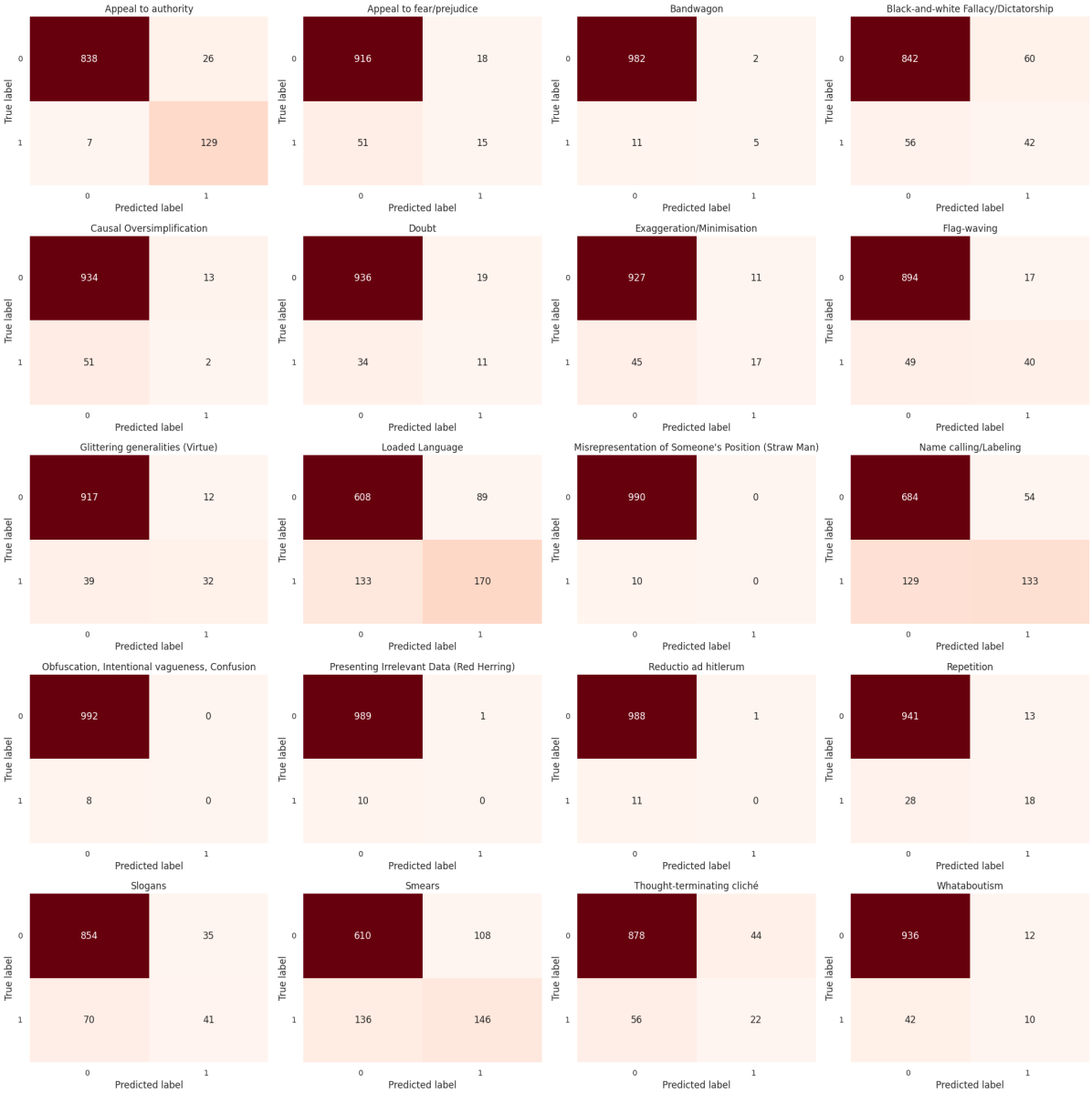


Figure 5: The confusion matrix of the model in the dev set of subtask 1.

Figure 6: The confusion matrix of the model in the dev set of subtask 2a.

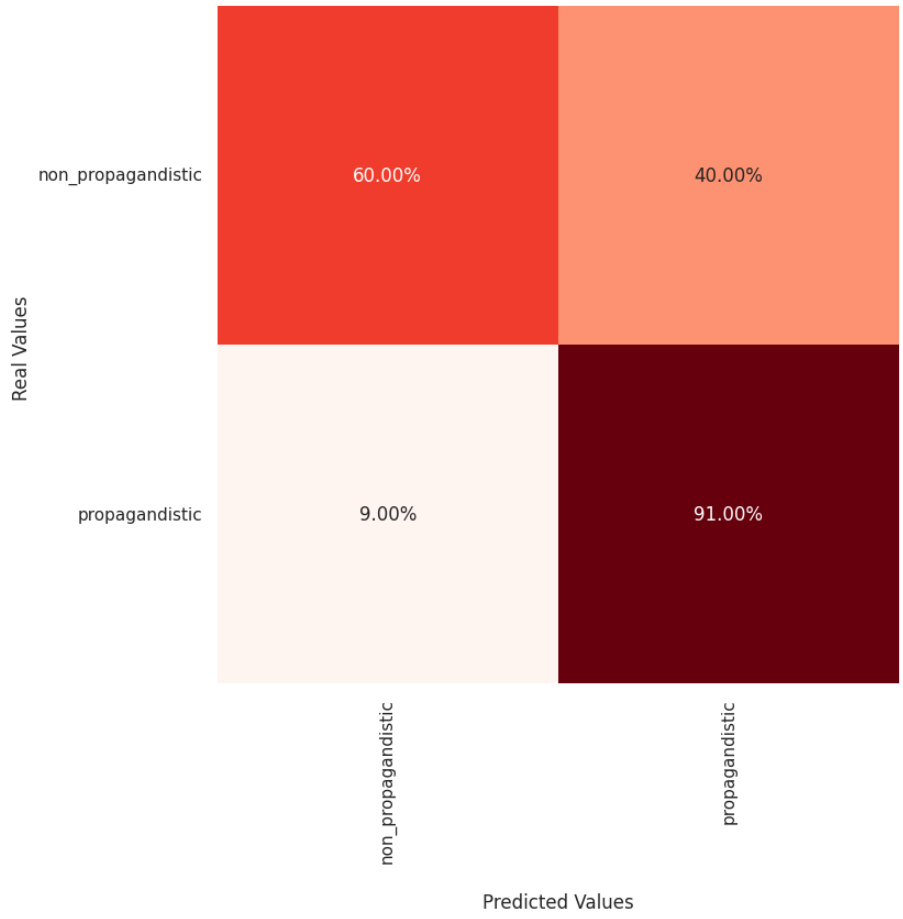


Figure 7: The confusion matrix of the model in the dev set of subtask 2b.

B Classification report with the dev set.

Table 5: Classification report of subtask 1 in the dev set.

	precision	**recall**	**f1-score**
Appeal to authority	83.2258	94.8529	88.6598
Appeal to fear/prejudice	45.4545	22.7273	30.3030
Bandwagon	71.4286	31.2500	43.4783
Black-and-white Fallacy/Dictatorship	41.1765	42.8571	42.0000
Causal Oversimplification	13.3333	03.7736	05.8824
Doubt	36.6667	24.4444	29.3333
Exaggeration/Minimisation	60.7143	27.4194	37.7778
Flag-waving	70.1754	44.9438	54.7945
Glittering generalities (Virtue)	72.7273	45.0704	55.6522
Loaded Language	65.6371	56.1056	60.4982
Misrepresentation of Someone's Position (Straw Man)	00.0000	00.0000	00.0000
Name calling/Labeling	71.1230	50.7634	59.2428
Obfuscation, Intentional vagueness, Confusion	00.0000	00.0000	00.0000
Presenting Irrelevant Data (Red Herring)	00.0000	00.0000	00.0000
Reductio ad hitlerum	00.0000	00.0000	00.0000
Repetition	58.0645	39.1304	46.7532
Slogans	53.9474	36.9369	43.8503
Smears	57.4803	51.7730	54.4776
Thought-terminating cliché	33.3333	28.2051	30.5556
Whataboutism	45.4545	19.2308	27.0270
micro avg	60.8918	46.0475	52.4394
macro avg	43.9971	30.9742	35.5143
weighted avg	58.2540	46.0475	50.6873
samples avg	46.3050	38.2436	39.3535

Table 6: Classification report of subtask 2a in the dev set.

	precision	**recall**	**f1-score**
Appeal to (Strong) Emotions	33.3333	17.8571	23.2558
Appeal to authority	83.1250	93.0070	87.7888
Appeal to fear/prejudice	46.4286	16.6667	24.5283
Bandwagon	75.0000	16.6667	27.2727
Black-and-white Fallacy/Dictatorship	38.6555	44.6602	41.4414
Causal Oversimplification	25.0000	03.5714	06.2500
Doubt	41.9355	25.0000	31.3253
Exaggeration/Minimisation	57.8947	16.1765	25.2874
Flag-waving	66.0000	53.6585	59.1928
Glittering generalities (Virtue)	63.0769	44.5652	52.2293
Loaded Language	70.7031	59.1503	64.4128
Misrepresentation of Someone's Position (Straw ...	00.0000	00.0000	00.0000
Name calling/Labeling	71.3592	56.3218	62.9550
Obfuscation, Intentional vagueness, Confusion	00.0000	00.0000	00.0000
Presenting Irrelevant Data (Red Herring)	00.0000	00.0000	00.0000
Reductio ad hitlerum	50.0000	12.5000	20.0000
Repetition	60.6061	43.4783	50.6329
Slogans	58.3333	42.6087	49.2462
Smears	72.7099	75.5952	74.1245
Thought-terminating cliché	28.7879	24.3590	26.3889
Transfer	59.3909	42.7007	49.6815
Whataboutism	60.0000	24.1935	34.4828
micro avg	64.7779	51.0870	57.1236
macro avg	48.2882	32.3971	36.8408
weighted avg	61.8299	51.0870	54.7241
samples avg	62.8705	52.9558	54.3457

Table 7: Classification report of subtask 2b in the dev set.

	precision	**recall**	**f1-score**
non_propagandistic	76.9231	60.0000	67.4157
propagandistic	81.9820	91.0000	86.2559
accuracy	80.6667	80.6667	80.6667
macro avg	79.4525	75.5000	76.8358
weighted avg	80.2957	80.6667	79.9759

MIPS at SemEval-2024 Task 3: Multimodal Emotion-Cause Pair Extraction in Conversations with Multimodal Language Models

Zebang Cheng[1*], **Fuqiang Niu**[1*], **Yuxiang Lin**[1]
Zhi-Qi Cheng[2], **Bowen Zhang**[1†], **Xiaojiang Peng**[1]
[1]Shenzhen Technology University [2]Carnegie Mellon University
zebang.cheng@gmail.com nfq729@gmail.com lin.yuxiang.contact@gmail.com zhiqic@cs.cmu.edu
zhang_bo_wen@foxmail.com pengxiaojiang@sztu.edu.cn

https://github.com/MIPS-COLT/MER-MCE.git

Abstract

This paper presents our winning submission to Subtask 2 of SemEval 2024 Task 3 on multimodal emotion cause analysis in conversations. We propose a novel Multimodal Emotion Recognition and Multimodal Emotion Cause Extraction (MER-MCE) framework that integrates text, audio, and visual modalities using specialized emotion encoders. Our approach sets itself apart from top-performing teams by leveraging modality-specific features for enhanced emotion understanding and causality inference. Experimental evaluation demonstrates the advantages of our multimodal approach, with our submission achieving a competitive weighted F1 score of 0.3435, ranking third with a margin of only 0.0339 behind the 1st team and 0.0025 behind the 2nd team.

1 Introduction

Emotion-Cause pair extraction in conversations has garnered significant attention due to its wide-ranging applications, such as optimizing customer service interactions and tailoring content recommendations based on user emotions (Xia and Ding, 2019). However, a fundamental challenge lies in identifying the causal determinants of emotional states. Recent research emphasizes the exploration of causes triggering emotions from multimodal data (Li et al., 2022), followed by further generation tasks based on multimodal emotional cues (Xu et al., 2024).

Practical conversations often exhibit multimodal cues through facial expressions, vocal changes, and textual content. Recognizing the significance of multimodal information, Wang et al. (2023) proposed the task of Multimodal Emotion-Cause Pair Extraction in Conversations (MECPE) as a critical step towards understanding the fundamental elicitors of emotions. The Emotion-Cause-in-

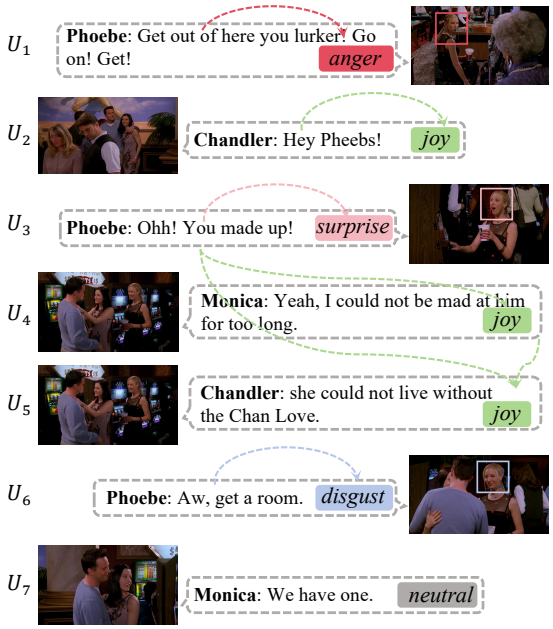


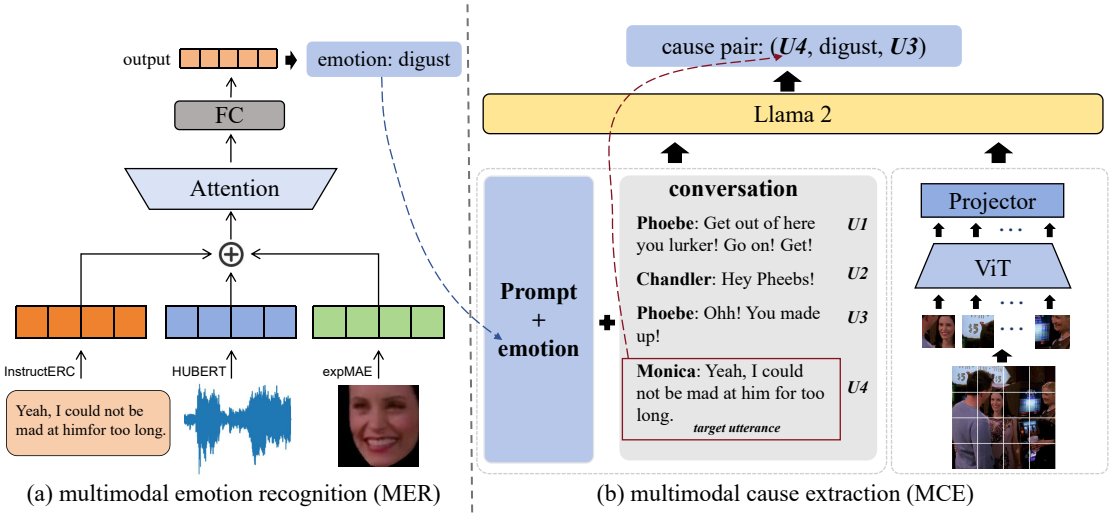
Figure 1: An example of an annotated conversation from the ECF dataset. Dashed lines connect each emotion label to its corresponding cause utterance, illustrating the emotion-cause pairs present in the conversation. The image modality provides additional context and cues for understanding the expressed emotions.

Friends (ECF) dataset, introduced in SemEval 2024 Task 3 (Wang et al., 2024), incorporates additional modalities such as images and audio alongside the original textual data, enabling a more realistic and comprehensive approach to emotion understanding. Figure 1 illustrates an example of an annotated conversation in the ECF dataset, where variations in facial expressions directly mirror the emotions expressed by the characters.

To address the MECPE task, we propose the Multimodal Emotion Recognition-Multimodal Cause Extraction (MER-MCE) framework, building upon the two-step approach introduced by Wang et al. (2023). Our method adopts a two-stage process to predict emotions and identify emotion causes in multimodal conversations. In the first stage, MER-MCE leverages text, audio, and image modalities

[*]Equal contributions, collaborated with CMU.
[†]Corresponding author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 667–674



(a) multimodal emotion recognition (MER) (b) multimodal cause extraction (MCE)

Figure 2: The architecture of our proposed MER-MCE framework for multimodal emotion-cause pair extraction in conversations. The framework consists of two main stages: (a) Multimodal Emotion Recognition (MER), which utilizes specialized emotion encoders to extract modality-specific features from text, audio, and visual data, and (b) Multimodal Cause Extraction (MCE), which employs a Multimodal Language Model to integrate contextual information from the conversation and visual cues to identify the utterances that trigger the recognized emotions.

for emotion prediction, utilizing state-of-the-art models specifically designed for capturing emotional cues from each modality. This approach sets our work apart from the first-place team in the competition, who relied solely on the textual modality for emotion recognition, and the second-place team, who employed a general-purpose model, ImageBind (Girdhar et al., 2023), for extracting visual and audio features.

In the second stage, considering the complexity of analyzing emotion causes for each utterance, we employ a Multimodal Large Language Model (LLM) to dissect the visual and textual modalities and discern the origins of each emotion. By leveraging the power of Multimodal LLMs, our approach can effectively capture the intricate relationships and dependencies present in real-world conversations, enabling a more nuanced and accurate identification of emotion causes.

The MER-MCE framework achieved notable results in Subtask 2 of SemEval 2024 Task 3, ranking third with a weighted F1 score of 0.3435, only 0.0339 behind the first-place team and 0.0025 behind the second-place team. We evaluate the two stages of our model separately to analyze their efficacy and limitations, providing valuable insights into the inherent challenges of the MECPE task. The main contributions of this paper are as follows:

- We propose the MER-MCE framework, a novel two-stage approach for Multimodal Emotion-Cause Pair Extraction in Conversations, leveraging state-of-the-art models for emotion recognition and Multimodal LLMs for cause extraction.
- We demonstrate the effectiveness of incorporating multiple modalities, including text, audio, and visual information, in both emotion recognition and cause extraction stages, leading to improved performance compared to approaches relying on a single modality or general-purpose feature extractors.
- Through comprehensive evaluation and analysis of the MER-MCE framework on the ECF dataset, we provide valuable insights into the challenges and opportunities in the field of multimodal emotion-cause pair extraction, paving the way for future research and advancements.

2 System Overview

In this work, we propose a Multimodal Emotion Recognition and Multimodal Cause Extraction (MER-MCE) framework for the task of multimodal emotion cause prediction and extraction (MECPE) in conversational settings. As illustrated in Figure 2, our MER-MCE model comprises two key modules: a multimodal emotion recognition (MER) module and a multimodal cause extraction (MCE) module, designed to work in tandem to tackle the intricate challenge of identifying emotions and their underlying causes from multimodal conversational data. Following we describe the structure of the entire system in detail.

2.1 Multimodal Emotion Recognition

Textual Modality. To comprehensively capture the rich semantic and contextual information present in real-world conversational content, our multimodal emotion recognition (MER) module adopts a carefully designed approach that leverages state-of-the-art models tailored for different modalities. For the textual modality, we employ the InstructionERC model proposed by Lei et al. (2023), which incorporates a domain demonstration recall module based on semantic similarity to enhance feature extraction. The textual features are extracted in the form of logits, capturing the nuanced semantic representations of the conversational utterances.

Auditory Modality. Recognizing the importance of auditory cues in conveying emotions, we utilize the HuBERT model proposed by Hsu et al. (2021) to process the audio modality and extract hidden states as acoustic features. These acoustic features encapsulate the rich tonal and prosodic information present in the audio signals, complementing the textual and visual modalities.

Visual Modality. For the visual modality, we first employ the OpenFace (Baltrusaitis et al., 2018) open-source tool to extract facial regions from video clips, allowing our visual model to focus specifically on facial expression recognition. Subsequently, we leverage the expMAE (Cheng et al., 2023) model to extract both static and dynamic features of facial expressions simultaneously. This dual-feature extraction approach captures the nuanced and time-varying aspects of facial expressions, which are known to be crucial indicators of emotional states.

Multimodal Fusion Mechanism. To effectively fuse the complementary information from these diverse modalities, we employ an attention-based multimodal fusion mechanism, as depicted in Figure 2(a). The input features from each modality are first mapped to a common 128-dimensional space, and ReLU activation and dropout regularization are applied to introduce non-linearity and improve generalization. We then compute attention weights via dot products between the features, allowing the model to dynamically attend to the most relevant and informative cues from each modality. The resulting fused representation captures the synergistic and complementary information across modalities, enabling a comprehensive understanding of the expressed emotions.

2.2 Multimodal Cause Extraction

Building upon the predicted emotions from the multimodal emotion recognition module, we propose a generative approach for multimodal cause extraction (MCE), harnessing the power of Multimodal Language Models (LLMs). As depicted in Figure 2(b), we adopt the MiniGPTv2 (Chen et al., 2023) model, a state-of-the-art multimodal LLM based on the LLaMA-2 (Touvron et al., 2023) architecture, as the backbone of our MCE module. This model is designed to integrate both visual and textual information, enabling it to extract emotional causes from multimodal conversational data.

Image Processing. The image processing component of our MCE module employs the Vision Transformer (ViT) (Dosovitskiy et al., 2020) model, which divides the input image into patches and extracts visual tokens representing these patches. Notably, in contrast to the image preprocessing approach used in the multimodal emotion recognition stage, we feed the complete image to the ViT encoder. This design choice allows our model to capture comprehensive scene information and relationships among individuals, providing a holistic visual context for emotion cause extraction. The extracted visual tokens are then mapped to the textual space using a linear projection layer, facilitating the seamless integration of visual and textual information within the multimodal LLM architecture.

Text Processing. To effectively incorporate contextual information from the conversation, we adopt a prompt-based approach in the textual processing component. The prompt template, illustrated in Figure 3, consists of two key elements: the prompt and the target utterance being queried. The prompt encompasses the conversation content preceding the target utterance, the speaker associated with the queried utterance, and the predicted emotion label obtained from the multimodal emotion recognition module.

Multimodal Cause Extraction. The integration of image and textual data is facilitated by a trainable LLaMA2-chat (7B) model, which processes the multimodal inputs and generates natural language responses to the posed inquiries. These responses are then subjected to a similarity matching process against utterances from the historical conversation dataset. This step allows us to identify the most relevant utterance that potentially triggered the recognized emotion, culminating in the extraction of the ultimate emotion cause utterance. By

seamlessly integrating visual and textual contextual information, our model can capture the intricate relationships and dependencies present in real-world conversations, enabling a more nuanced and flexible representation of emotion causes. This generative approach paves the way for more accurate and interpretable emotion cause extraction, a critical component in the overall task of multimodal emotion cause prediction and extraction.

prompt: Now you are expert of sentiment and emotional analysis. The following conversation noted between [***conversation***] involves several speakers.
Please infer, considering the conversation content and the facial expressions of the characters in the image, which utterance leads to [***speaker***] feeling [***emotion***].
target: [***conversation*** U_i].

Figure 3: Prompt template for guiding the Multimodal LLM in sentiment analysis and emotion cause extraction from conversational data.

3 Experiments

3.1 Experimental Setup

We evaluate our MER-MCE model on the Emotion-Cause-in-Friends (ECF) dataset (Poria et al., 2018), an extension of the multimodal MELD dataset that includes emotion cause annotations in addition to the original emotion labels. The dataset provides annotations for utterances that trigger the occurrence of emotions, enabling the study of emotion-cause pairs in conversations. For Subtask 2, the labeled data is divided into "train," "dev," and "test" subsets, containing 1001, 112, and 261 conversations, respectively.

Our experimental setup involves extracting features from textual, audio, and visual modalities using various state-of-the-art pretrained models. For the textual modality, we use models such as XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), BERT (Devlin et al., 2018), and InstructERC (Lei et al., 2023) to capture semantic and contextual information. For the audio modality, we employ models like VGGish (Hershey et al., 2017), wav2vec (Schneider et al., 2019), and HUBERT (Hsu et al., 2021) to extract features that encapsulate tonal and prosodic information. In the case of the visual modality, we use pretrained visual models, such as MANet (Zhao et al., 2021), ResNet (He et al., 2016), and expMAE (Cheng et al., 2023), to capture nuanced and time-varying aspects of facial expressions conveying emotional information.

3.2 Evaluation Metrics

The primary objective of Subtask 2 is to predict emotion-cause pairs for non-neutral categories based on the provided conversations. The performance of the participating models is evaluated using a weighted average of the F1 scores across six emotion categories: anger, disgust, fear, joy, sadness, and surprise. This weighted average F1 score provides a comprehensive evaluation of the model's ability to accurately predict emotion-cause pairs while considering the imbalanced nature of the dataset.

Further details on the dataset, experimental setup, and evaluation metrics can be found in the supplementary material.

3.3 Emotion Recognition Analysis

We conducted an extensive experimental evaluation of the Multimodal Emotion Recognition (MER) component within the MER-MCE framework. Table 1 presents the weighted F1 scores for various state-of-the-art models evaluated in the emotion recognition task, leveraging features extracted from textual, audio, and visual modalities.

Our analysis reveals that the textual modality, which captures rich semantic information conveyed through conversations, plays a crucial role in emotion recognition. The inherent ability of the textual modality to encapsulate abstract semantic information facilitates the extraction of emotional features, resulting in higher scores compared to other modalities in unimodal emotion recognition.

To exploit the complementary nature of different modalities, we performed multimodal feature fusion. The experimental results (Table 1) demonstrate that increasing the number of modalities leads to a significant improvement in emotion recognition accuracy, empowering the model to effectively discern emotions within more complex samples. Notably, even features that exhibit relatively lower precision in unimodal emotion recognition contribute positively when integrated into the multimodal fusion framework, highlighting the importance of multimodal approaches in capturing fine-grained emotional nuances.

However, our analysis also uncovers challenges in the visual and audio modalities. In sitcoms containing multiple characters, the OpenFace tool struggles to accurately identify the current speaker, leading to noise in the visual features. Similarly, canned laughter from the audience contributes to

T	A	V	w-avg. F1
XLNet	-	-	0.4418
RoBERTa	-	-	0.5036
BERT	-	-	0.5128
InstructERC	-	-	**0.6606**
-	VGGish	-	0.2657
-	wav2vec	-	0.4021
-	**HUBERT**	-	**0.4403**
-	-	MANet	0.3999
-	-	ResNet	0.4035
-	-	**expMAE**	**0.4104**
InstructERC	VGGish	-	0.6729
InstructERC	HUBERT	-	0.6749
InstructERC	-	ResNet	0.6774
InstructERC	-	expMAE	0.6781
-	HUBERT	ResNet	0.5113
-	HUBERT	expMAE	0.5099
InstructERC	VGGish	ResNet	0.6758
InstructERC	VGGish	expMAE	0.6779
InstructERC	HUBERT	ResNet	0.6792
InstructERC	**HUBERT**	**expMAE**	**0.6807**

Table 1: Multimodal emotion recognition results

noise in the audio modality. Consequently, models trained on these modalities exhibit subpar performance compared to the textual modality.

3.4 Cause Extraction Analysis

In the MCE stage, we conducted a comparison of the cause extraction capabilities between different models and the state-of-the-art MECPE-2steps model (Wang et al., 2023), with the test results presented in Table 2. Initially, we employed the same attention model (Lian et al., 2023) structure as in MER for cause extraction. However, this relatively simple model struggled to capture the relationships between utterances. We then explored the ALBEF model (Li et al., 2021) based on the transformer structure, which allowed the model to focus on the connections between different utterances. Nevertheless, limited training data and imbalanced data distribution led to overfitting.

To address these challenges, we transformed the emotion cause extraction task from a traditional discriminative architecture to a generative architecture based on Multimodal LLM, resulting in improved cause extraction accuracy. We utilized the historical conversation window in the Multimodal LLM prompt to retain contextual information within the conversation. Ablation experiments were conducted to investigate the impact of varying numbers of historical conversation windows on cause extraction (Figure 4). To accurately assess the true influence of MCE, we employed the

Method	F1	w-avg. F1
Heuristic	-	0.1864
MECPE-2steps	-	0.3315
Attention	0.3415	0.3403
ALBEF	0.3644	0.3672
MER-MCE(ours)	**0.4074**	**0.4042**

Table 2: Multimodal cause extraction results

actual labels from the test dataset for cause extraction instead of relying solely on the emotions predicted in MER. Our experiments revealed that the effectiveness was maximized when the number of historical windows reached 5. However, as the number of windows increased further, the effectiveness gradually decreased due to the complexity of conversations with a larger historical context.

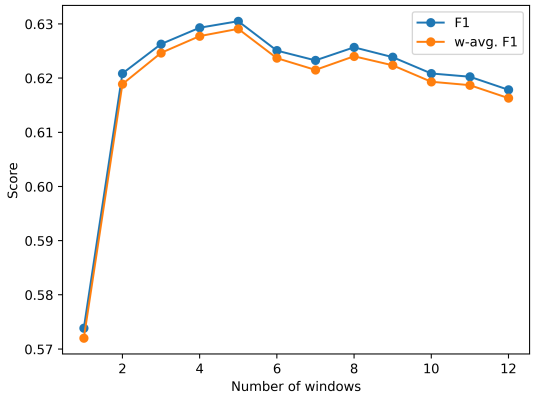


Figure 4: The line graph depicting scores and historical conversation windows.

3.5 Error Analysis of the Entire System

We conducted quantitative and qualitative error analysis on the two stages of our MER-MCE framework to identify key limitations.

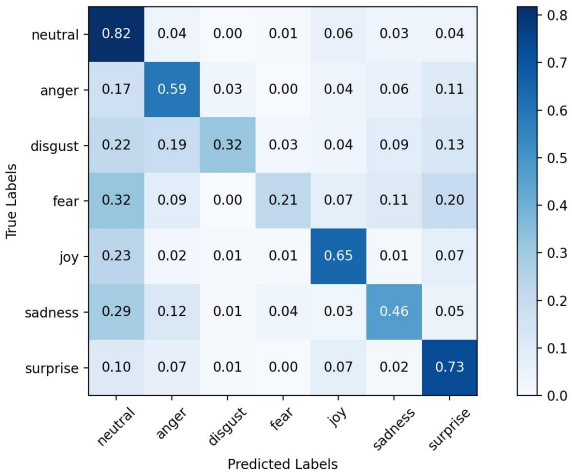


Figure 5: The confusion matrix of multimodal emotion recognition result.

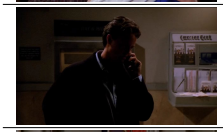
Visual Modality	Historical Conversation Content	Label Pair	Pred Pair
	U4: Cat. U5: Yes! You are so smart! I love you. U6: I love you too.	target: U6 emotion: joy cause: U5	target: U6 emotion: joy cause: U5
	U1: I have no idea what you just said. U2: Put Joey on the phone.	target: U2 emotion: anger cause: U1	target: U2 emotion: anger cause: None
	U2: You know what? It really creeps me out ... U3: Sorry. U4: I am so exited!	target: U4 emotion: joy cause: U5	target: U4 emotion: joy cause: U4
	U9: Sure. Okay. U10: Uh , are you crazy? Are you insane? ... U11: Yeah, I ..., I just know it would make me happy.	target: U11 emotion: neutral cause: None	target: U11 emotion: joy cause: U11

Table 3: Analysis of typical predicted emotion-cause pairs generated by the model, with emphasis on samples labeled as 'neutral' that do not have an associated emotion cause.

Analysis of the emotion recognition results using a confusion matrix (Figure 5) revealed that approximately 20% of non-neutral emotion categories were misclassified as neutral, hindering the subsequent cause analysis stage and impacting the overall recall rate. Additionally, class imbalance in the dataset adversely affected the performance of the "disgust" and "fear" categories, which had the lowest number of annotations.

Further analysis of the final Emotion-Cause Pair, based on representative samples in Table 3, highlighted the impact of different scenarios on our model. Facial occlusion in the visual modality (second sample) led to erroneous emotion classification, suggesting the need for more robust visual processing techniques. Strong emotional distractors in the textual modality (fourth sample) misled the model, emphasizing the importance of sophisticated language understanding methods to disambiguate distractors effectively. The real-time setting posed challenges in capturing long-range dependencies and identifying causes in future utterances (third sample), indicating the need for techniques that can model long-range dependencies and incorporate future context.

Despite these challenges, our MER-MCE model demonstrated the ability to accurately predict emotion-cause pairs by leveraging contextual information from both visual and textual modalities (first sample in Table 3). It identified key areas for improvement, including handling facial occlusion, disambiguating emotional distractors, and capturing long-range dependencies in real-time settings.

4 Conclusion

This paper introduces the MER-MCE model for emotion cause analysis in conversations, developed for SemEval 2024 Task 3. Our model leverages multimodal information and Language Models (LLMs) to identify emotion causes in conversational data, considering textual, visual, and audio modalities. MER-MCE achieves a weighted F1 score of 0.3435, ranking third in the competition. The results demonstrate the effectiveness of multimodal approaches in capturing emotional dynamics. Future work will focus on enhancing generalizability and robustness by exploring additional modalities and advanced techniques. We plan to investigate the incorporation of pose estimagesture recognition and facial expression analysis to improve the model's ability to detect emotions.

5 Acknowledgements

This work was supported by the National Natural Science Foundation of China (grants 62306184 and 62176165), the Stable Support Projects for Shenzhen Higher Education Institutions (grant 20220718110918001), and the Natural Science Foundation of Top Talent of SZTU (grants GDRC202320 and GDRC202131). Zhi-Qi Cheng acknowledges support from the Air Force Research Laboratory (agreement FA8750-19-2-0200), the Defense Advanced Research Projects Agency (DARPA) grants funded through the GAILA program (award HR00111990063), and the AIDA program (FA8750-18-20018). We also appreciate the organizers of SemEval 2024.

References

Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 59–66. IEEE.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Zebang Cheng, Yuxiang Lin, Zhaoru Chen, Xiang Li, Shuyi Mao, Fan Zhang, Daijun Ding, Bowen Zhang, and Xiaojiang Peng. 2023. Semi-supervised multimodal emotion recognition with expression mae. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9436–9440.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework. *arXiv preprint arXiv:2309.11911*.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

Wei Li, Yang Li, Vlad Pandelea, Mengshi Ge, Luyao Zhu, and Erik Cambria. 2022. Ecpec: emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*.

Zheng Lian, Haiyang Sun, Licai Sun, Jinming Zhao, Ye Liu, Bin Liu, Jiangyan Yi, Meng Wang, Erik Cambria, Guoying Zhao, et al. 2023. Mer 2023: Multi-label learning, modality robustness, and semi-supervised learning. *arXiv preprint arXiv:2304.08981*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Chao Xu, Yang Liu, Jiazheng Xing, Weida Wang, Mingze Sun, Jun Dan, Tianxin Huang, Siyuan Li, Zhi-Qi Cheng, Ying Tai, et al. 2024. Facechain-imagineid: Freely crafting high-fidelity diverse talking faces from disentangled audio. *arXiv preprint arXiv:2403.01901*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zengqun Zhao, Qingshan Liu, and Shanmin Wang. 2021. Learning deep global multi-scale and local attention features for facial expression recognition in the wild. *IEEE Transactions on Image Processing*, 30:6544–6556.

A Appendix

A.1 Experimental Data

The ECF dataset, an extension of the multimodal MELD dataset (Poria et al., 2018), includes emotion cause annotations in addition to the original emotion labels. It provides annotations for utterances that trigger the occurrence of emotions, enabling the study of emotion-cause pairs in conversations.

For Subtask 2, the labeled data is divided into "train," "dev," and "test" subsets, containing 1001, 112, and 261 conversations, respectively. We followed this partition to train, validate, and test our MER-MCE model. To evaluate results, we submitted predictions for an additional 655 unlabeled conversations to the CodaLab platform[1].

A.2 Experimental Setup

To evaluate the effectiveness of our MER-MCE model for the MECPE task, we employed various state-of-the-art pretrained models to extract features from the textual, audio, and visual modalities.

For the textual modality, we directly used the conversation text from the annotated files as input to models such as XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), BERT (Devlin et al., 2018), and InstructERC (Lei et al., 2023) to extract textual features that capture semantic and contextual information.

For the audio modality, we extracted audio files from the video clips using FFMPEG[2] and fed them into models like VGGish (Hershey et al., 2017), wav2vec (Schneider et al., 2019), and HUBERT (Hsu et al., 2021) to obtain audio features that encapsulate tonal and prosodic information.

In the case of the visual modality, we used the OpenFace toolkit to extract facial features from the video clips while masking out the background. We then employed pretrained visual models, such as MANet (Zhao et al., 2021), ResNet (He et al., 2016), and expMAE (Cheng et al., 2023), which were initially trained on facial expression datasets, to extract visual features that capture nuanced and time-varying aspects of facial expressions conveying emotional information.

By leveraging these diverse pretrained models across multiple modalities, we aim to comprehensively capture the rich emotional cues present in the conversations and evaluate the effectiveness of our MER-MCE model in integrating these multimodal features for emotion-cause pair extraction.

A.3 Evaluation Metrics

The primary objective of Subtask 2 is to predict emotion-cause pairs for non-neutral categories based on the provided conversations. Each emotion-cause pair ($p_i = eu_i, ec_i, cu_i$) consists of three essential elements: the index of the emotion utterance eu_i, the emotion category ec_i, and the index of the cause utterance cu_i.

The performance of the participating models is evaluated using a weighted average of the F1 scores across the six emotion categories: anger, disgust, fear, joy, sadness, and surprise. The F1 score for each emotion category j is computed as follows:

$$F_1^j = \frac{2 \times precision^j \times recall^j}{precision^j + recall^j},$$

where $precision^j$ and $recall^j$ are the precision and recall scores for emotion category j, respectively.

The overall performance is determined by the weighted average F1 score, which takes into account the number of samples in each emotion category:

$$w\text{-}avg.F_1 = \frac{\sum_{j=1}^{6} n^j \times F_1^j}{\sum_{j=1}^{6} n^j},$$

where n^j denotes the number of samples of category j. This weighted average F1 score provides a comprehensive evaluation of the model's ability to accurately predict emotion-cause pairs across different emotion categories while considering the imbalanced nature of the dataset.

[1] https://codalab.lisn.upsaclay.fr/competitions/16141
[2] https://ffmpeg.org/

UMUTeam at SemEval-2024 Task 6: Leveraging Zero-Shot Learning for Detecting Hallucinations and Related Observable Overgeneration Mistakes

Ronghao Pan[1], **José Antonio García-Díaz**[1], **Tomás Bernal-Beltrán**[1], **Rafael Valencia-García**[1]

[1] Facultad de Informática, Universidad de Murcia, Campus de Espinardo, 30100 Murcia, Spain

{ronghao.pan, joseantonio.garcia8, tomas.bernalb, valencia}@um.es

Abstract

In these working notes we describe the UMUTeam's participation in SemEval-2024 shared task 6, which aims at detecting grammatically correct output of Natural Language Generation with incorrect semantic information in two different setups: model-aware and model-agnostic tracks. The task is consists of three subtasks with different model setups. Our approach is based on exploiting the zero-shot classification capability of the Large Language Models LLaMa-2, Tulu and Mistral, through prompt engineering. Our system ranked eighteenth in the model-aware setup with an accuracy of 78.4% and 29th in the model-agnostic setup with an accuracy of 76.9333%.

1 Introduction

Recently, the emergence of Large Language Models (LLMs) has brought about a paradigm shift in Natural Language Processing (NLP), leading to unprecedented advances in Natural Language Understanding (NLU) (Huang et al., 2023) and Reasoning (Zhang et al., 2023). In general, LLMs refer to a set of general-purpose models based on the Transformer architecture and pre-trained on large text corpora, such as GPT-3 (Brown et al., 2020), LLaMa (Touvron et al., 2023), PaLM (Chowdhery et al., 2023) and GPT-4 (Achiam et al., 2023). By scaling the amount of data and model capacity, LLMs demonstrate incredible emergent capabilities, typically including In-Context Learning (ICL) (Brown et al., 2020), chain-of-thought prompting (Wei et al., 2022), and instruction following (Peng et al., 2023).

Natural Language Generation (NLG) faces two related challenges. First, current models often produce output that is fluent but inaccurate. Second, the metrics used to evaluate the LLMs performance prioritize fluency over correctness, exacerbating the problem of "hallucination", in which LLMs produce fluent but incorrect output. Consequently, significant research in underway to automatically detect such errors. In many NLG applications, output correctness is paramount, as in cases such as machine translation, where producing a plausible but inconsistent translation compromises the utility of the system.

Thus, the SHROOM shared-task focuses on identifying grammatically correct outputs that contain incorrect semantic information, regardless of whether the model producing the output is accessible or not (Mickus et al., 2024). To this end, the organizers have adapted a post-hoc environment in which the models have already been trained, and the outputs have already been produced. The participants' task is a binary classification problem to identify cases of hallucinations, i.e. to detect grammatically correct outputs that contain incorrect or unsupported semantic content, in two different setups: model-aware and model-agnostic tracks.

To address the SHROOM challenge, our team used a zero-shot learning (ZSL) approach with LLaMa-2 (Touvron et al., 2023), Mistral (Jiang et al., 2023), and Tulu (Ivison et al., 2023) LLMs to detect grammatically correct output that contains incorrect semantic information through the prompt. The ZSL technique refers to the ability of LLMs to perform tasks without being explicitly trained on them, meaning that the model can generate responses or make predictions on topics or domains that were not part of its explicit training. This is achieved by exploiting the general knowledge that the models have acquired during their massive pretraining on large text corpora.

During our experiments, we observed that these LLMs were able to identify hallucinations. In particular, Tulu is the one best suited for this task.

The rest of this paper is organized as follows. Section 2 provides a summary of important details about the task setup. Section 2 gives an overview of our system for two subtasks. Section 4 presents the specific details of our systems. Section 5 dis-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 675–681

cusses the results of the experiments, and finally, the conclusions are presented in Section 6.

2 Background

NLG is a branch of Artificial Intelligence (AI) and computational linguistics that deals with the automated generation of text in human language. NLG covers a wide range of tasks, such as text generation for chatbots, automatic summarization, machine translation, story generation, and others. NLG relies on models and algorithms that enable machines to understand and generate text that is coherent and intelligible to humans. However, current models can produce inaccurate but fluent output, while the metrics tend to describe fluency rather than correctness. This leads to models producing "hallucinations", i.e. generated content that appears nonsensical or unfaithful to the given source content.

In general, hallucinations in NLG tasks can be divided into two main types (Ji et al., 2023): intrinsic hallucinations and extrinsic hallucinations. On the one hand, intrinsic hallucinations refer to the output of LLMs that conflict with the source content. On the other hand, extrinsic hallucinations refer to LLM generations that cannot be verified from the source content.

The SHROOM task aims to automatically detect hallucinations and related observable overgeneration errors. To achieve this, the organizers have provided a collection of checkpoints, inputs, references and outputs from systems covering three different NLG tasks: (1) definition modeling (DM), (2) machine translation (MT), and (3) paraphrase generation (PG), trained with different levels of accuracy. The development set includes binary annotations from at least five different annotators and a majority vote gold label.

The generalizability of LLMs is very attractive because it allows us to adapt state-of-the-art methods to specific goals. For example, an LLM trained on multilingual texts can perform translations without being explicitly trained to do so (known as zero-shot capability, ZSL). Another possibility is to guide models by providing them with examples of the input and the desired output (known as few-shot learning, FSL). For example, in (García-Díaz et al., 2023), LLMs have shown good performance in a ZSL scenario for identifying hate speech in Spanish and English. In this sense, it is possible to ask for a sentence and its translation before asking it to translate another sentence. This additional information helps to improve the quality of the output. For text classification tasks, the ability to make such predictions with little or no training makes these models particularly promising for empirical research, as they have the potential to perform accurately without the need for costly and time-consuming annotation procedures.

Therefore, we took advantage of this ZSL classification capability of LLMs to detect hallucinations and related observable overgeneration errors.

The following models are evaluated:

- **Mistral** (Jiang et al., 2023). Higher model performance often requires an escalation in model size. However, this scalability tends to increase computational cost and inference latency, raising the barriers to implementation in practical real-world scenarios. Mistral 7B is a high-performance LLM that maintains efficient inference. Mistral 7B outperforms the 13 billion parameter LLaMa-2 model on all benchmarks. In addition, Mistral 7B approaches the coding performance of Code-Llama 7B without sacrificing performance on non-code benchmarks.

- **LLaMa-2** (Touvron et al., 2023). Llama 2 and Llama 2-Chat are pre-trained and fine-tuned LLMs, both at scales of up to 70B parameters. In several benchmarks tested, Llama 2-Chat models generally outperformed existing open-source models. For our system, we used an instructively fine-tuned version of LLaMa-2 with 7B parameters from the Orca (Mukherjee et al., 2023) set called "stabilityai/StableBeluga-7B[1]".

- **Tulu** (Ivison et al., 2023). TuLu is a family of pre-trained and fine-tuned LLMs. Unlike other existing LLMs, distilled data mixtures from TuLu have been shown to significantly improve downstream performance over instruction and datasets available, with a new mixture outperforming its predecessor by an average of 8%. In addition, TuLu models use a fine-tuned version of Direct Preference Optimization (DPO) that scales to 70 billion parameter models and significantly improves open-response generation metrics without compromising model performance, im-

[1] https://huggingface.co/stabilityai/StableBeluga-7B

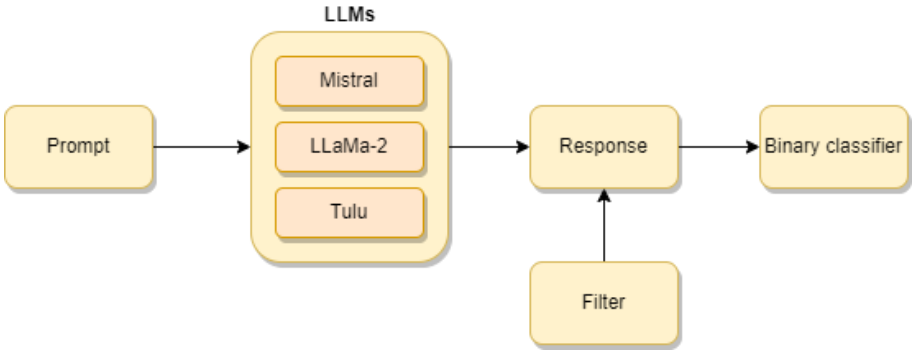


Figure 1: System architecture approach

proving AlpacaEval performance by an average of 13% across all model scales. For this task, we have evaluated the 7 billion parameter DPO version of Tulu with called "tulu-2-dpo-7b".[2]

3 System overview

Figure 1 shows the architecture of our system. We can see that we have introduced a specific prompt for each LLM to generate a response with the desired structure. Then we have a module called "filter" that extracts a binary response based on the response and the correlation value.

3.1 Prompt

The prompt in the context of LLMs refers to a specific input provided to the model to elicit a desired response or to guide the text generation. This prompt can be a sentence, a question, or even a fragment of text that sets the context or direction for the model's text generation. In our proposal, we use prompt engineering, which involves the design and careful wording of these prompts to elicit specific model responses and optimally influence the model's response.

Figure 2 shows the prompts used for each LLM, in which we can see that each LLM has its own control tokens to indicate which parts are system control sequences and which parts are user questions. For example, in LLaMa-2 "`### System`" is used to indicate the control sequence, and "`### User`" is used to indicate the user question. However, Mistral and Tulu do not have tokens to indicate system control sequences, but we can append the control sequence to the user question.

In our system, we have used the same prompt structure for all LLMs: (1) **System control sequence**. It is used to specify the context and instruct the model to respond only with "yes" or "no" at the beginning of the response; and (2) **User question**. It is used to introduce the text and specify the question "the Sentence supported by the Context above". Once the response generated by the LLMs is obtained, it is passed through the filter module, which identifies the first word of the response and classifies it as "Hallucination" or "Not Hallucination". To obtain the correlation value, we have used the same approach as the baseline provided by the organizers, which consists of extracting the log probability value of the first token of the response generated by the LLM.

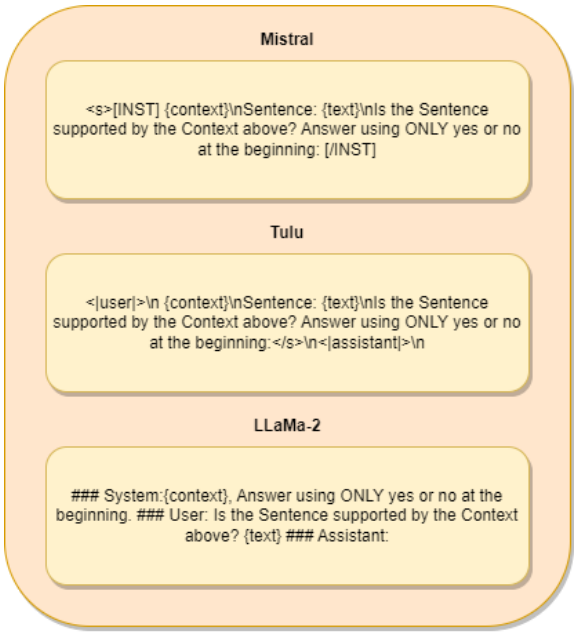


Figure 2: Examples of prompts for each LLMs

4 Experimental setup

In this section, we explain the dataset used, the hyperparameters used in the LLMs to generate re-

[2] https://huggingface.co/allenai/tulu-2-dpo-7b

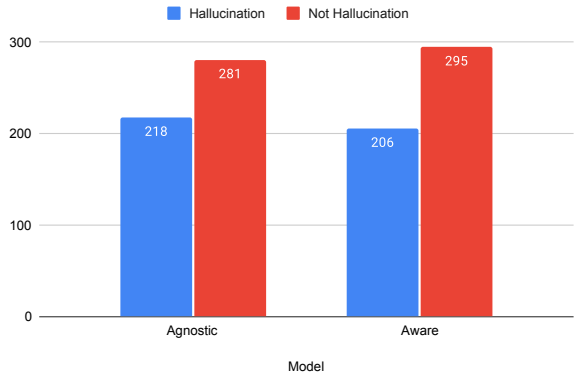


Figure 3: Validation set distribution

sponses, and details of the metrics used by the organizers for evaluation.

In this task, the organizers provided participants with unlabeled train data, trial data, and validation data for both the model-aware and model-agnostic setups. We have only used the validation data to evaluate the performance of different models using a ZSL approach. Figure 3 displays the distribution of the validation set.

The hyperparameters used in the LLMs to generate the response are 0.95 for top_p, 0 for top_k, 256 for max_new_tokens and the default temperature for each LLM.

The evaluation metrics used are accuracy for binary classification and rho to evaluate correlation. The rho metric, commonly known as the rho correlation coefficient (ρ), is a statistical measure that evaluates the relationship between two ordinal variables. It is particularly useful when the variables are not continuous but are divided into ordered categories.

5 Results

Table 1 shows the results of different LLMs in the validation set with two different configurations: (1) model-aware and (2) model-agnostic tracks. Thus, the system has to identify when a text is grammatically correct but contains incorrect information inconsistent with the source input, either with or without access to the model that produced the text. We can see that the Tulu performed best in both the model-aware and model-agnostic configurations. It obtained an accuracy of 76.6467% and a rho of 0.521104 in the model-aware configuration and an accuracy of 73.7475% and a rho of 0.553962 in the model-agnostic metric.

According to the results with development, we used Tulu in the task. Table 2 shows the official ranking for the task. We achieved the eighteenth best result out of a total of 46 teams in the model-aware setup, with a precision of 78.4% and a rho of 0.506895. Compared to result to the best result, our model is 2.866% worse in precision and 19.25% worse in rho. Regarding the model independent setup, our system achieved the nineteenth best result out of 49 participants, with a precision of 76.9333%, which is 7.8% worse than the best team (ahoblitz), and a rho of 0.560945, which is 20.86% worse than the best team.

Table 1: Results obtained with different LLMs in the validation set.

LLM	Accuracy	Rho
Aware		
Mistral	52.2954	0.345239
Tulu	**76.6467**	**0.521104**
LLaMa-2	66.8663	0.487483
Agnostic		
Mistral	50.3006	0.229504
Tulu	**73.7475**	**0.553962**
LLaMa-2	65.5310	0.521414

Table 2: Official raking table

LLM	Rank	Accuracy	Rho
Aware			
HaRMoNEE	1	81.2666	0.699316
ahoblitz	2	80.6000	0.714712
TU Wien	3	80.6000	0.707192
	...		
UMUTeam	18	**78.4000**	**0.506895**
Agnostic			
ahoblitz	1	84.7333	0.769512
OPDAI	2	83.6000	0.732195
HIT_WL	3	83.0666	0.767700
	...		
UMUTeam	29	**76.9333**	**0.560945**

5.1 Error analysis

We perform an error analysis of our system. For this, we extracted the confusion matrix from Tulu on the test set of the two configurations (model-aware and model-agnostic).

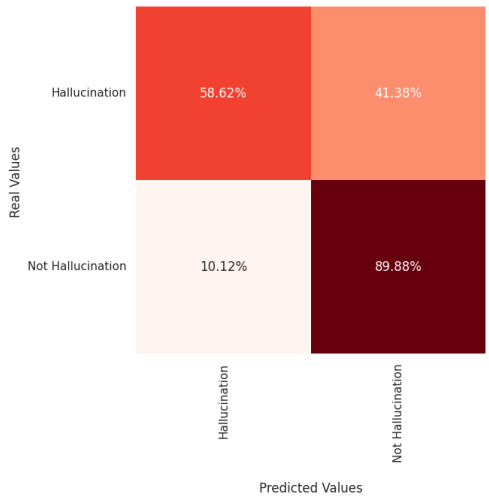


Figure 4: Confusion matrix of Tulu with test dataset in model-aware setup.

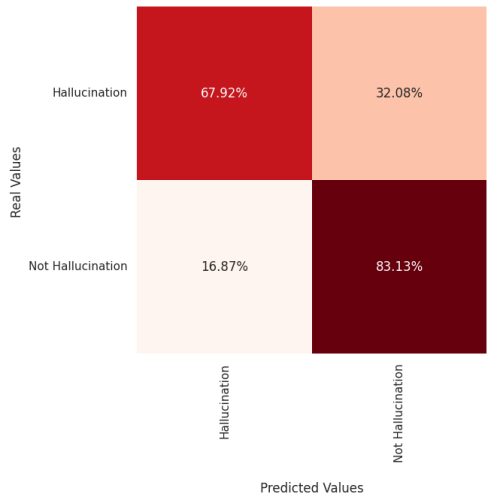


Figure 5: Confusion matrix of Tulu with test dataset in model-agnostic setup.

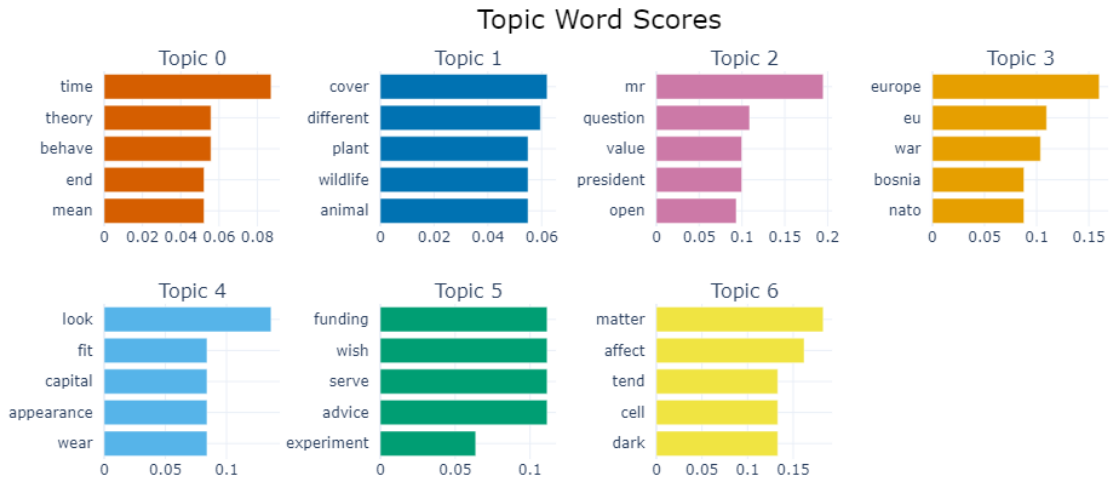


Figure 6: The most frequent topics associated with misclassification in model-aware setup.

Figure 4 shows the confusion matrix of TuLu from the test set in the model-aware setup. This approach tends to confuse hallucinations with non-hallucinations with a probability of 41.38%. However, it performs very well at detecting “not hallucination” with a probability of 89.88%. Regarding the model-agnostic setup, our model tends to confuse hallucinations with "not hallucination" with a probability of 32.08%, but is able to identify “not hallucination” with an accuracy of 83.13%.

The Tulu model from the test set in the model-aware setup has obtained a total of 324 misclassifications, of which 165 are of the definition modeling type, 100 are of the machine translation type, and 59 are of the paraphrase generation type. Therefore, we have a total of 165 misclassifications with the Flan-T5[3] model, 100 with the NLLB[4] model, and 59 with the Pegasus Paraphrase[5] model. In order to know the most common topic that the model comments on the classification error, we used the BERTopic model to identify and group topics in the context of the failed cases. In Figure 6 we can see the 7 topics in which the TuLu model usually misidentifies.

Regarding the model-agnostic setup, our ap-

[3] ltg/flan-t5-definition-en-base
[4] facebook/nllb-200-distilled-600M
[5] tuner007/pegasus_paraphrase

proach has obtained a total of 346 misclassifications, of which 138 are of the definition modeling type, 107 are of the machine translation type, and 101 are of the paraphrase generation type. In contrast to the model-aware setup, there is an increase in the accuracy of the identification of definition modeling misclassifications, but a decrease in the identification of paraphrase generation misclassifications. Figure 7 shows the three most common topics associated with the classification errors.

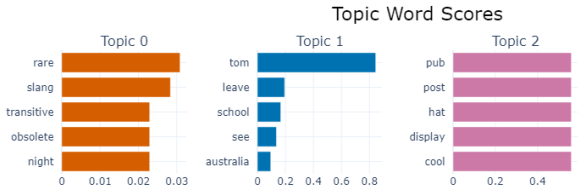


Figure 7: The most frequent topics associated with misclassification in model-agnostic setup.

6 Conclusion

Here we describe the UMUTeam's participation in SHROOM (SemEval 2024), concerning the development of models for detecting grammatically correct output from NLGs, but with incorrect semantic information in two different setups: model-aware and model-agnostic tracks. We have used the ZSL approach with LLaMa-2 (Touvron et al., 2023), Mistral (Jiang et al., 2023), and Tulu (Ivison et al., 2023) LLMs to detect output that contains incorrect semantic information through the prompt. Tulu performed best in the evaluation set. Using this model, we ranked eighteenth in the model-aware setup with an accuracy of 78.4% and nineteenth in the model-agnostic setup with an accuracy of 76.9333%.

As further work, we propose to investigate hallucination detection in the political domain. In politics, automated content generation can help politicians to generate text on a variety of political topics, which can help political campaigns, think tanks, and government agencies quickly produce tailored content. Hallucination detection can help to mitigate misleading or fabricated content. In this sense, we propose to generate political discourse that imitates politicians from different political wings (García-Díaz et al., 2022) and to identify the generated hallucinations by different LLMs.

Acknowledgements

This work is part of the research projects LaTe4PoliticES (PID2022-138099OB-I00) funded by MICIU/AEI/10.13039/501100011033 and the European Regional Development Fund (ERDF)-a way to make Europe and LT-SWM (TED2021-131167B-I00) funded by MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. In addition, this work was funded by the Spanish Government, the Spanish Ministry of Economy and Digital Transformation through the Digital Transformation through the "Recovery, Transformation and Resilience Plan" and also funded by the European Union NextGenerationEU/PRTR through the research project 2021/C005/0015007. Mr. Ronghao Pan is supported by the "Programa Investigo" grant, funded by the Region of Murcia, the Spanish Ministry of Labour and Social Economy and the European Union - NextGenerationEU under the "Plan de Recuperación, Transformación y Resiliencia (PRTR)".

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

José Antonio García-Díaz, Salud M Jiménez Zafra, María Teresa Martín Valdivia, Francisco García-Sánchez, Luis Alfonso Ureña López, and Rafael Valencia García. 2022. Overview of politices 2022: Spanish author profiling for political ideology. *Procesamiento del Lenguje Natural*.

José Antonio García-Díaz, Ronghao Pan, and Rafael Valencia-García. 2023. Leveraging zero and few-shot learning for enhanced model generality in hate speech detection in spanish and english. *Mathematics*, 11(24):5004.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

DFKI-NLP at SemEval-2024 Task 2: Towards Robust LLMs Using Data Perturbations and MinMax Training

Bhuvanesh Verma
DFKI GmbH
Universität Potsdam
bhuvanesh.verma@dfki.de

Lisa Raithel
BIFOLD
Quality & Usability Lab, TU Berlin
DFKI GmbH
Université Paris-Saclay, CNRS, LISN
raithel@tu-berlin.de

Abstract

The NLI4CT task at SemEval-2024 emphasizes the development of robust models for Natural Language Inference on Clinical Trial Reports (CTRs) using large language models (LLMs). This edition introduces interventions specifically targeting the numerical, vocabulary, and semantic aspects of CTRs. Our proposed system harnesses the capabilities of the state-of-the-art Mistral model (Jiang et al., 2023), complemented by an auxiliary model, to focus on the intricate input space of the NLI4CT dataset. Through the incorporation of numerical and acronym-based perturbations to the data, we train a robust system capable of handling both semantic-altering and numerical contradiction interventions. Our analysis on the dataset sheds light on the challenging sections of the CTRs for reasoning.

1 Introduction

Over the last decade, Natural Language Processing (NLP) has seen significant advancements, beginning with the introduction of word embeddings (Mikolov et al., 2013), followed by transformer architectures like BERT (Vaswani et al., 2017; Devlin et al., 2019), and specialized language models (LMs) such as BioBERT (Lee et al., 2020) and PubMedBERT (Gu et al., 2021) tailored for the biomedical domain. The advent of large language models (LLMs) like GPT-3 (Brown et al., 2020), commonly known as Chat-GPT, has further pushed the boundaries of NLP, showcasing capabilities in diverse NLP tasks and even reasoning. However, LLMs adapt to shortcut learning easily instead of understanding the task at hand and resorting to shallow lexical heuristics for making a prediction (Tsuchiya, 2018; Poliak et al., 2018; Naik et al., 2018). Additionally, we have seen generative models like Chat-GPT hallucinating, making false claims, and struggling with providing factual information (Elazar et al., 2021; Wang et al., 2023). Tackling these challenges is essential for ensuring the reliable deployment of large language models, particularly in critical fields like biomedicine, where the margin for error must be minimized.

The SemEval-2024 Task 2: *Safe Biomedical Natural Language Inference for Clinical Trials* is focused on improving the understanding and evaluation methodologies for Large Language Models in clinical Natural Language Inference (NLI) (Jullien et al., 2024). This task targets aspects such as numerical and quantitative reasoning, domain-specific terminology, syntax, and semantics. It aims to analyze models' robustness, consistency, and faithfulness in reasoning within the clinical domain.

Our approach to this task involved leveraging instruction fine-tuned LLMs along with an auxiliary model that focuses on "hard" instances to develop a more resilient NLI system. "Hard" instances refer to those examples in the dataset where the model fails. Building on the methodology outlined by Kanakarajan and Sankarasubbu (2023), we assessed the zero-shot performance of various instruction-tuned LLMs to identify the most effective model. Upon selecting the best LLM, we introduced an auxiliary module during the fine-tuning process, which emphasized learning "hard" examples. Taking inspiration from Korakakis and Vlachos (2023), who experimented with various configurations for the auxiliary module and highlighted its substantial impact on the final NLI system's performance, we explored various architectures for this auxiliary module. To improve the robustness of the system and address challenges related to numerical reasoning and domain-specific terminology, we introduced numerical and semantic perturbation to the NLI4CT dataset and trained our system on these. Our system ranked 11th in *macro* F_1 *score*, 12th in *Faithfulness*, and 19th *Consistency* out of 31 participants. Our final system struggled when dealing with semantic-preserving interventions on

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 682–696

the test data yet demonstrated strong performance on semantic-altering interventions.

2 Background

We now provide a description of the shared task, followed by a brief overview of the NLI4CT dataset. We then explore existing research, assessing their strengths and limitations while also drawing connections to our proposed method.

2.1 Task and Dataset Description

This task is a continuation from SemEval-2023 Task 7 (Valentino et al., 2023), which introduced the NLI4CT dataset (Jullien et al., 2023) derived from Clinical Trial Reports (CTRs) on breast cancer. The dataset contains 999 CTRs, each of which consists of four sections: `Eligibility Criteria`, a set of conditions for patients to be allowed to take part in the clinical trial; `Intervention`, information concerning the type, dosage, frequency, and duration of treatments being studied; `Results`, the number of participants in the trial, outcome measures, units, and the results; and `Adverse Events`, signs and symptoms observed in patients during the clinical trial. The dataset comprises two types of training instances: *single* and *comparison*. In the *single* instances, one section of the CTR serves as the premise, while a corresponding human-annotated statement is presented as the hypothesis. On the other hand, in the *comparison* instances, the same section of two CTRs is utilized, and the hypothesis typically involves a human-annotated comparative statement between the two CTRs. Each instance is labeled either **entailment** or **contradiction**, with an equal distribution of proportions between the two labels (more details in Appendix A.1). A sample instance for *single* is shown in Figure 1.

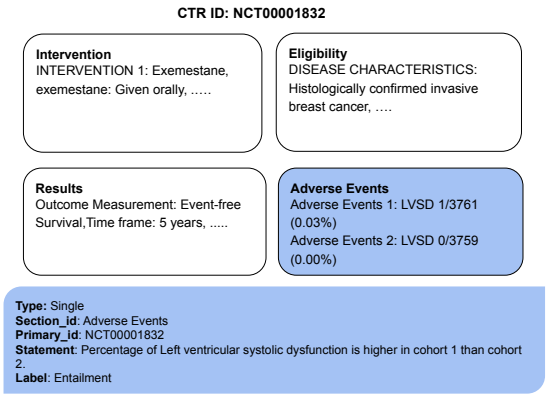


Figure 1: A sample instance from the NLI4CT dataset. Each instance consists of four sections: `Intervention`, `Eligibility criteria`, `Results`, and `Adverse Events`. The data are split into two types: *single* (depicted) and *comparison*. In *single*, one section of the CTR serves as the premise (in this case, `Adverse Events`). A human-annotated hypothesis for this premise is given (Statement), which is then to be classified into either **entailment** or **contradiction**.

2.2 Related Works

The NLI4CT dataset (Jullien et al., 2023) was introduced in SemEval 2023 Task 7 (Valentino et al., 2023), where multiple submissions highlighted the aforementioned challenges associated with language models. The second-ranked team from SemEval 2023 Task 7, Saama AI Research (Kanakarajan and Sankarasubbu, 2023), initially evaluated an instruction-tuned LLM in a zero-shot setting. Subsequently, they fine-tuned the model using the best instruction with T5 (Raffel et al., 2020) and Flan-T5-XXL (Chung et al., 2022). Motivated by their methodology and drawing inspiration from recent advancements, we employed instruction-tuned LLMs such as Llama (Touvron et al., 2023) and Mistral (Jiang et al., 2023), which represent state-of-the-art LLMs. Additionally, building upon the work of Korakakis and Vlachos (2023), who introduced a learner-auxiliary model framework to enhance the robustness of NLI, we aimed to integrate this framework alongside the use of instruction-tuned LLMs in our approach.

The challenge of word distribution shift from the general domain to the biomedical domain has posed a significant obstacle to the effectiveness of NLP methods applied to the biomedical field. The prevalence of aliases and acronyms in biomedical text prompted Jin et al. (2019) to propose a model that automatically collects context for abbreviations from PubMed abstracts and employs a BiLSTM classifier for abbreviation expansion. Additionally, Grossman Liu et al. (2021) presented a Medical Abbreviation and Acronym Meta-Inventory[1], constituting a comprehensive database of medical abbreviations encompassing 104,057 entries, each linked to 170,426 corresponding senses. We leveraged this Meta-Inventory to incorporate acronym-based perturbations into the NLI4CT dataset. Additionally, we also incorporated a pre-finetuning phase

[1] https://github.com/lisavirginia/clinical-abbreviations

into our approach by fine-tuning on the MedNLI[2] dataset (Shivade et al., 2019). This step aims to familiarize the model with clinical data.

3 System Overview

In light of recent advancements in large language models and drawing insights from the results of the SemEval 2023 Task 7 (Valentino et al., 2023), we implemented the approach outlined in the work of Kanakarajan and Sankarasubbu (2023). Our approach involved evaluating state-of-the-art LLMs, including Mistral (Jiang et al., 2023), Llama (Touvron et al., 2023), and Lemma (Azerbayev et al., 2024), alongside their variants. We experimented with different instructions for each model and subsequently compared their zero-shot performance based on their respective best-performing instruction (see final instruction template in Appendix A.3). Mistral emerged as the top-performing model among all others evaluated with the highest F_1 score (0.69). Furthermore, we implemented the MinMax algorithm (Korakakis and Vlachos, 2023) by adding an auxiliary model alongside the Mistral model to create a more robust system. This auxiliary model is designed to amplify the loss incurred in input spaces where the Mistral model encounters difficulties, effectively directing its focus towards areas of higher loss. To further boost the performance of the system, we pre-finetuned using MedNLI dataset. Additionally, we conducted an error analysis to identify easy and difficult instances in the train set to provide a basis for further research.

4 Experimental Setup

Training an LLM can be both costly and resource-intensive. However, recent advancements in methodologies, such as Parameter-Efficient Fine-Tuning (PEFT), have emerged to reduce the computational cost of fine-tuning (Mangrulkar et al., 2022). For fine-tuning the Mistral model, we employ a PEFT method known as Low-Rank Adaption (LoRA, Hu et al. (2022)). We adopted a similar approach for implementing the auxiliary model as described by Korakakis and Vlachos (2023). We experimented with the parameters of the system to obtain an optimal architecture, details of which can be found in Appendix A.5.4.

[2] https://physionet.org/content/mednli/1.0.0/

4.1 Data Perturbation

Utilizing the Meta-Inventory of Grossman Liu et al. (2021), we extracted the short forms from 358 NLI4CT hypotheses, resolving them to their corresponding long forms based on the cosine similarity. This resulted in 352 perturbed instances with consistent labels. Additionally, 181 negative instances were generated by selecting the least similar long forms, resulting in a total of 533 new instances for the acronym-based perturbation. For numerical perturbations, we employed an English Named Entity Recognition model (Raza et al., 2022) trained on Maccrobat to extract 27 unique biomedical entities from hypotheses. We perturbed numerical values and introduced semantic alterations that generated 355 new instances with labels flipped. For more details, see Appendix A.4.

4.2 Fine-tuning Strategies

We performed various experiments involving different combinations of fine-tuning methodologies. Initially, we fine-tuned only the Mistral model (*NLI4CT-FT*) on NLI4CT without incorporating the auxiliary model. An extension of this initial setup involved N-step fine-tuning, where, for example, in a two-step fine-tuning approach, we first fine-tuned the model with the MedNLI dataset and subsequently fine-tuned it further with the NLI4CT dataset (*MEDNLI-FT-NLI4CT*). We proceeded to add more steps by fine-tuning on perturbated datasets, such as the acronym-perturbed dataset (*MEDNLI-NLI4CT-FT-ACR*) or the numerically-perturbed dataset (*MEDNLI-NLI4CT-FT-NUM*). The MinMax algorithm requires that the base model be trained for a few epochs or steps. We utilized the best-performing models from previous N-step experiments to adapt this strategy effectively. This way, we already have a model that is trained on the dataset and add the auxiliary model to enhance the robustness of the whole system. Details of all the models that we fine-tuned with different strategies can be found Appendix A.2.

4.3 Evaluation Strategies

In our initial experiments, we observed that Mistral 7B exhibited superior performance compared to Mistral Instruct 7B post-fine-tuning. Consequently, we primarily trained most models using Mistral 7B. However, during the evaluation phase, we attached the PEFT fine-tuned adapter with both Mistral and Mistral Instruct 7B to compare their results. To sta-

Model	Dev F_1	Test F_1	Consistency	Faithfulness
NLI4CT-FT	0.69	0.74	0.68	0.75
NLI4CT-FT-ACR	0.73	0.76	0.67	0.71
MEDNLI-FT-NLI4CT	0.75	0.75	0.68	0.78
MEDNLI-FT-NLI4CT-ACR-NUM	0.75	0.74	0.68	0.78
MEDNLI-NLI4CT-FT-ACR	0.74	0.75	0.67	0.74
MEDNLI-NLI4CT-FT-NUM	0.74	0.73	0.69	0.79
MEDNLI-NLI4CT-FT-ACR-NUM	0.75	**0.76**	**0.70**	0.75
MINMAX-MEDNLI-FT-NLI4CT	0.75	0.75	0.68	**0.82**
MINMAX-MEDNLI-FT-NLI4CT-BC	**0.77**	0.75	0.68	0.78
MINMAX-MEDNLI-NLI4CT-FT-ACR-NUM-BC	0.74	0.74	0.68	0.75

Table 1: Final results on the NLI4CT dataset. **Dev** F_1 and **Test** F_1 represent the *macro* F_1 score on the development and test set, respectively. **Consistency** measures the ability to predict same labels for *semantic preserving* interventions and **Faithfulness** measures the ability to correctly change the labels for *semantic altering* interventions. Both *Consistency* and *Faithfulness* results are on the test set.

bilize the model's generation behavior, we conduct evaluations on the development set five times and select the label predicted most frequently across these five runs. Similarly, for test data, we perform three runs.

5 Results

During both the fine-tuning and evaluation phases, we observed improvements in the model trained with the MinMax algorithm compared to other models. From Table 1, we can see the model (*MINMAX-MEDNLI-FT-NLI4CT-BC*) trained with the MinMax algorithm achieved the highest F_1 score on the dev set. When comparing the base model (*MEDNLI-FT-NLI4CT*) with the MinMax model (*MINMAX-MEDNLI-FT-NLI4CT*), we noted a slight improvement in *Consistency* and a significant improvement in *Faithfulness*. Although the F_1 score did not exhibit improvement, the enhancements in the other metrics indicate that the MinMax algorithm contributed to the development of a more robust system and was able to handle the semantically altering intervention much better. Regarding the models trained with perturbed data, we observed a negative effect on the overall performance of the MinMax-trained model (*MINMAX-MEDNLI-NLI4CT-FT-ACR-NUM-BC*) compared to the base model (*MEDNLI-NLI4CT-FT-ACR-NUM*). For our final submission to the leaderboard, we submitted the MinMax model (*MINMAX-MEDNLI-FT-NLI4CT*), which ranked 11th in *macro* F_1 *score*, 12th in *Faithfulness*, and 19th in *Consistency*.

5.1 Impact of Data Perturbation

To assess the impact of acronym-based perturbed data, we initially trained the model using the original NLI4CT dataset and subsequently fine-tuned it with the acronym-based data. Evaluation of both models was conducted on the test data, which comprise the following intervention types introduced by the task's organizers: Control, Contrast, Paraphrase, Contradiction, Numerical Contradiction, Numerical Paraphrase, and Definitions. For accessing a model trained on acronym-based perturbed data, we look at the metrics for the intervention types Paraphrase (*Para*) and Definitions. Table 4 in Appendix A.4.3 shows that acronym perturbation notably enhanced results for the intervention-type Definitions. Similarly, with numerical-based perturbation, we look at metrics for intervention-type Numerical Paraphrase (*Num_Para*) and Numerical Contradiction (*Num_Cont*). While no changes were observed in the results of semantic-altering interventions, there was some improvement noted in semantic-preserving interventions. Lastly, we investigated the combined impact of both perturbations and their influence on all four interventions. Overall, we observed that combined fine-tuning improved the Definition and Paraphrase intervention type more than the Numerical intervention types. However, there was a negative impact observed on semantic-altering numerical interventions.

5.2 Performance across Interventions and Sections

Table 5 in Appendix A.6 presents the results of the test data across various interventions and sections. The `Adverse Events` section exhibits the highest F_1 score at 0.73, whereas the `Eligibility` section demonstrates the lowest score at 0.66. In terms of interventions, Numerical_contradiction achieves the highest score at 0.93, while Definition attains the lowest at 0.63. Among the interventions fea-

turing both labels, Paraphrase achieves the highest performance with an F_1 score of 0.72. Moreover, it is the only intervention type that achieves a higher score for **entailment**. Conversely, all other sections and interventions exhibit better performance for the **contradiction** label.

6 Error Analysis

To understand the model's behavior across different sections, interventions, and labels/relations, we examined the dataset.

6.1 Dataset Difficulty

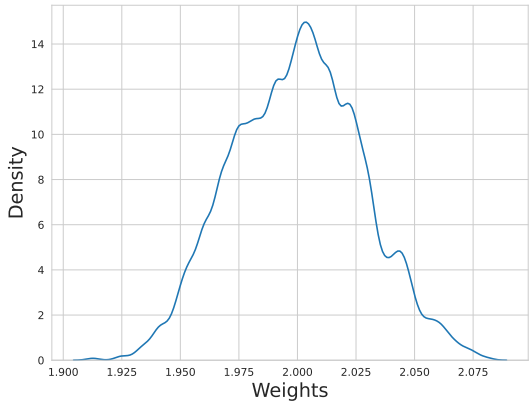


Figure 2: Weight distribution of NLI4CT data instances generated by the auxiliary model after 3 epochs of training. Lower weights correspond to easy examples, and higher weights correspond to hard examples.

One application of the MinMax algorithm is its capability to classify data points into hard and easy examples. Figure 2 illustrates the weight distribution of data instances from the auxiliary model after three epochs of training. Data instances with higher weights represent hard examples, where the model incurs a high loss, while instances with lower weights denote easy examples.

Following the data cartography procedure outlined in Swayamdipta et al. (2020), we replicated their method using the best MinMax model trained for three epochs. We collected probability values for the gold label on each epoch and calculated confidence, variability, and correctness values. In Figure 3, the upper region with red data points represents easy-to-learn instances, while the bottom region with blue data points represents hard-to-learn examples. Data points with high variability are depicted as ambiguous examples.

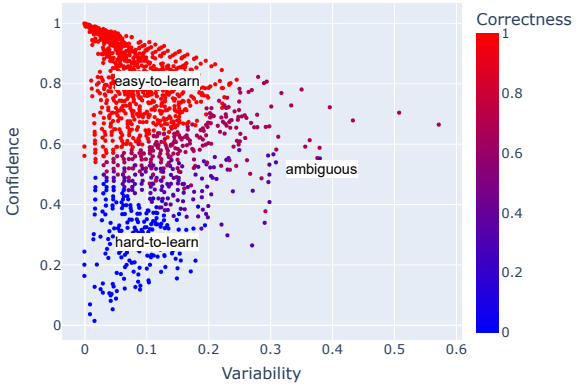


Figure 3: Data map for the NLI4CT dataset following (Swayamdipta et al., 2020).

6.2 Analysis of Easy and Hard Samples

We conducted a comparison between the two dataset difficulty methodologies by extracting easy and hard examples from both strategies. We found 322 instances common to both strategies as easy examples or easy-to-learn instances. As for hard examples or hard-to-learn examples, there were 96 instances common to both. We performed a three-level analysis using these instances, especially the hardest ones, to understand the in-depth dataset difficulty and the model's behavior.

First, we looked at the structural level of the dataset concerning these instances and found that instances focusing on the `Eligibility` section were identified as the most easy-to-learn for the model, whereas those targeting `Adverse Events` proved challenging. Additionally, learning the **contradiction** relation was more difficult than **entailment** (see Table 7). Next, we compared the word overlap between the premise and hypothesis of the easy and hard examples. We found that the word overlap was higher in the easy examples compared to the hard examples. Furthermore, the easy examples exhibited a higher frequency of **entailment** relations, suggesting that the model might have established a correlation between word overlap and **entailment** relations (see Figure 4). One potential solution to mitigate this issue could involve perturbing the instances with high word overlap by introducing synonyms into the dataset.

Combining observations from these analyses provides some interesting insights. As previously discussed in 5.2, the results from the test data reveal that the `Eligibility` section obtained the lowest F_1 score, while `Adverse Events` performed the

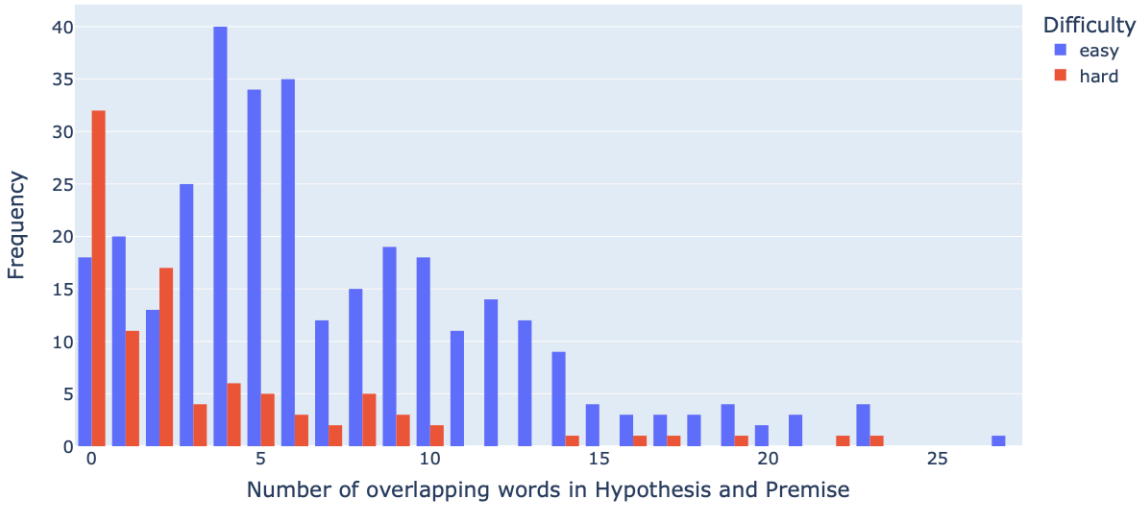


Figure 4: Word overlap between the hypothesis and the premise in the easy and the hard examples.

best. Given that the instances of the `Eligibility` section in the training set were easy to learn, it is plausible that the model did not learn many features from this section. Conversely, as the instances of `Adverse Events` were more challenging to learn, the model likely attempted to extract more features from this section. A similar rationale can also be applied to the **entailment** and **contradiction** relation. However, another factor contributing to the higher scores on the **contradiction** relations in the test data could be the greater number of the true **contradiction** relations.

Finally, we manually analyzed the ten most difficult examples. We discovered that the predominant error made by the model involved the confusion between the cohorts and the trials. Specifically, the instances that involve a comparison between two trials, each comprising two cohorts, often led the model to misinterpret the second cohort of the first trial as the secondary trial. Overall, the model struggled with numerical reasoning, particularly in scenarios involving numerous variables that require calculations. More details on the analysis of dataset difficulty can be found in Appendix A.8.

7 Conclusion

In this study, we introduced a large language model-based system designed to address the natural language inference task through text generation. Our approach prioritized model robustness, which was achieved by incorporating an auxiliary model that directs the LLM to focus on challenging instances in the input space. Moreover, we enhanced the system's robustness against adversarial samples by introducing numerical and semantic perturbations to the NLI4CT dataset during training. Our findings revealed the system's superior robustness against semantic-altering interventions compared to semantic-preserving ones. Additionally, through dataset analysis, we identified instances targeting the `Eligibility` section in Clinical Trial Reports as the the easiest to learn but more challenging to predict accurately. Conversely, the `Adverse Events` section posed greater difficulty in learning but was relatively easier to predict accurately. These findings offer valuable insights for future research on improving the robustness by focusing more on challenging sections of CTRs.

Acknowledgments

Our work was funded by the Deutsche Forschungsgemeinschaft (German Research Foundation) DFG-442445488 under the trilateral ANR-DFG-JST AI research project KEEPHA. Furthermore, we gratefully acknowledge funding from the German Federal Ministry of Education and Research under the grant BIFOLD24B.

Limitations and Ethical Considerations

We do not rule out the possible risk of sensitive content in the data. Furthermore, the Mistral-based models in our experiments, which were pre-trained on a wide variety of source data, might have inherited biases from these pretraining corpora. We

further acknowledge that prompts used to generate responses with Mistral models might result in different responses when the prompts are slightly modified or set up differently.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *URL https://openai.com/blog/sparse-transformers*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Lisa Grossman Liu, Raymond H Grossman, Elliot G Mitchell, Chunhua Weng, Karthik Natarajan, George Hripcsak, and David K Vawdrey. 2021. A deep database of medical abbreviations and acronyms for natural language processing. *Scientific Data*, 8(1):149.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Joy He-Yueya, Gabriel Poesia, Rose E Wang, and Noah D Goodman. 2023. Solving math word problems by combining language models with symbolic solvers. *arXiv preprint arXiv:2304.09102*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Qiao Jin, Jinling Liu, and Xinghua Lu. 2019. Deep contextualized biomedical abbreviation expansion. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 88–96.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama ai research at semeval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003.

Michalis Korakakis and Andreas Vlachos. 2023. Improving the robustness of nli models with minimax training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14322–14339.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language

representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Shaina Raza, Deepak John Reji, Femi Shajan, and Syed Raza Bashir. 2022. Large-scale application of named entity recognition to biomedicine and epidemiology. *PLOS Digital Health*, 1(12):e0000152.

Chaitanya Shivade et al. 2019. Mednli-a natural language inference dataset for the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium. Association for Computational Linguistics*, pages 1586–1596.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, et al. 2023. Semeval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Jie Yao, Zihao Zhou, and Qiufeng Wang. 2023. Solving math word problem with problem type classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 123–134. Springer.

A Appendix

A.1 Dataset Statistics

We highlight the basic statistics of the NLI4CT dataset in Table 2.

A.2 Descriptions of the Fine-tuned Models

We implemented various fine-tuning strategies across multiple models. Below, we provide descriptions for each of these models:

- **NLI4CT-FT**: Mistral-7B model fine-tuned on the NLI4CT dataset.
- **MEDNLI-FT**: Mistral-7B model fine-tuned on the MEDNLI dataset.
- **NLI4CT-FT-ACR**: NLI4CT-FT model fine-tuned on acronym based perturbations.
- **MEDNLI-FT-NLI4CT**: MEDNLI-FT fine-tuned on the NLI4CT dataset.
- **MEDNLI-FT-NLI4CT-ACR-NUM**: MEDNLI-FT fine-tuned simultaneously on the NLI4CT dataset, acronym and numerical perturbations.
- **MEDNLI-NLI4CT-FT-ACR**: MEDNLI-FT-NLI4CT model fine-tuned on acronym perturbations.
- **MEDNLI-NLI4CT-FT-NUM**: MEDNLI-FT-NLI4CT model fine-tuned on numerical perturbations.
- **MEDNLI-NLI4CT-FT-ACR-NUM**: MEDNLI-FT-NLI4CT model fine-tuned on acronym and numerical perturbations simultaneously.
- **MINMAX-MEDNLI-FT-NLI4CT**: MEDNLI-FT fine-tuned using Mistral-7B and the auxiliary model on the NLI4CT dataset.
- **MINMAX-MEDNLI-FT-NLI4CT-BC**: MEDNLI-FT fine-tuned using Mistral-7B and the auxiliary model on the NLI4CT dataset using the best configuration obtained from hyperparameter tuning.
- **MINMAX-MEDNLI-NLI4CT-FT-ACR-NUM-BC**: MINMAX-MEDNLI-FT-NLI4CT fine-tuned using Mistral-7B and the auxiliary model on acronym and numerical perturbations simultaneously using the best configuration obtained from hyperparameter tuning.

```
<s>### Instruction:
Read the input text and answer the
following question with Yes or No.

### Input:
{premise}

Question: Does this imply that
{hypothesis}?

### Response:
{label}</s>
```

Figure 5: Final design for prompting.

A.3 Final Instruction Template

After running experiments with different prompt formats, we finalized the template as shown in Figure 5. Instead of directly tackling the NLI task, we frame it as a text generation problem. We begin by giving general instructions, which describe the task to be performed. The next two sections of the prompt consist of the premise, providing context for the task, and the hypothesis presented as a question. The model is then trained to generate either "Yes" for an **entailment** relationship between the premise and hypothesis or "No" for a **contradiction**. While fine-tuning our model with the MedNLI dataset, we only utilized entailment and contradiction instances, excluding those labeled as neutral, to ensure consistency with the NLI4CT dataset.

A.4 Data Perturbation Details

In Table 3, we show full statistics of data perturbation on NLI4CT dataset. In the following section, we describe the data perturbation methodology.

A.4.1 Acronym Based Perturbations

We utilized a Medical Abbreviation and Acronym Meta-Inventory (Grossman Liu et al., 2021) containing short forms (SF) and corresponding long forms (LF) commonly used in the biomedical domain. With regular expressions, we extracted short forms present in the hypotheses of the NLI4CT dataset, resulting in 358 hypotheses. Given that the meta-inventory often includes multiple long

Data	No of Samples	Type	Section				Label	
			Count	Intervention	Eligibility	Adverse Events	Results	contradiction	entailment
Train	1700	*single*	1035	155	317	309	254	502	533
		comparison	665	241	169	187	68	348	317
Dev	200	*single*	140	26	44	32	38	70	70
		comparison	60	10	12	20	18	30	30
Test	5500	*single*	2553	784	468	523	778	1703	850
		comparison	2947	758	951	781	457	1956	991

Table 2: NLI4CT statistics.

Data	No of Samples	Type	Section				Label	
			Count	Intervention	Eligibility	Adverse Events	Results	contradiction	entailment
ACR	533	*single*	357	67	103	20	167	178	179
		comparison	176	46	70	53	7	93	83
NUM	355	*Single*	268	51	67	39	111	267	1
		comparison	87	24	25	31	7	86	1

Table 3: Statistics for Acronym the (ACR) and Numerical (NUM) based perturbed dataset across different sections, labels, and instance types.

forms for a single short form we computed the cosine similarity between the short forms in the hypotheses and their corresponding long forms in the meta-inventory. For each unique short form identified in the hypotheses, we determined the most similar long form and manually verified its correctness within the context of the hypothesis. Subsequently, we resolved the short forms in the format: 'SF (LF)'. This process yielded 352 perturbed instances with consistent inference labels. Such perturbations are intended to assist models in avoiding potential confusion by ensuring that short forms are resolved, even when their corresponding long forms are present in the premise. Likewise, for each unique short form, we identified the least similar long form and generated a negative instance following the same format as before. This process resulted in approximately 181 new negative instances, where labels were flipped. Consequently, when combining both acronym-based perturbations, we created a total of 533 new instances.

A.4.2 Numerical Perturbation

The Math Word Problem (MWP) task has been introduced in NLP to enhance models' numerical reasoning capabilities (He-Yueya et al., 2023; Yao et al., 2023). Within our dataset, numerous instances involve comparisons of numerical entities, which inherently qualify as MWPs. To augment these instances, we introduce noise to the numerical entities in various forms. Utilizing an English Named Entity Recognition model (Raza et al., 2022) trained on Maccrobat, specifically tailored for biomedical entities (107 entities), we extracted 27 unique entities from the hypotheses. Our focus was on identifying entities that can alter the hypothesis's meaning concerning numerical reasoning, such as Age, Dosage, Lab_value, Duration, and Date. For numerical values associated with these entities, we applied basic mathematical operations like addition or subtraction. Additionally, words comparing these numerical entities were replaced with their opposites; for example, 'lower' was substituted with 'higher', and 'more than a week' was replaced with 'less than a week', and so forth. This process resulted in a total of 355 new perturbed instances, each with its label flipped.

A.4.3 Data Perturbations Results

Table 4 presents results for various interventions introduced in test data. *Base* in the table refers to *MEDNLI-FT-NLI4CT*.

A.5 Model and Experiment Details

We provide information regarding the models, the minmax algorithm, and experiments.

A.5.1 Mistral 7B and Mistral Instruct 7B

Mistral 7B, as the name suggests, has 7 billion parameters and stands out as a language model engineered for exceptional performance and efficiency. Central to its architecture are the grouped-query attention (Ainslie et al., 2023) and sliding window attention mechanisms (Child et al., 2019; Beltagy et al., 2020). Mistral models demonstrate remark-

Model	F_1				Faithfulness		Consistency	
	Def	Para	Num_Para	Num_Cont	Def	Para	Num_Para	Num_Cont
Base	0.42	0.72	0.54	0.88	0.59	0.72	0.68	0.90
Base + ACR	0.49	0.73	**0.59**	0.82	0.61	0.71	0.68	0.88
Base + NUM	0.46	0.73	0.56	**0.88**	0.60	0.72	0.68	**0.91**
Base + ACR + NUM	**0.58**	**0.73**	0.58	0.83	**0.64**	**0.72**	**0.68**	0.90
MinMax + ACR + NUM	0.51	0.73	0.56	0.83	0.62	0.71	0.68	0.88

Table 4: Acronym (ACR) and Numerical (NUM) perturbed dataset results

able adaptability and consistently outperform counterparts like Llama-13B. Moreover, the ease with which Mistral can be fine-tuned is evidenced by the Mistral Instruct 7B version, which is fine-tuned on publicly available instruction datasets and achieves a significant performance boost over the base version. Utilizing the capabilities of Mistral models, we fine-tuned both versions of the models on NLI4CT through a series of experiments aimed at determining the optimal version for final system development. Details of the Mistral models are shown in Table 6.

A.5.2 Low Rank Adaption

LoRA operates by freezing the weights of the pretrained model and introducing trainable rank decomposition matrices into each layer of the Transformer architecture. This strategy significantly reduces the number of trainable parameters for downstream tasks, leading to lower memory usage and accelerated fine-tuning speed. We utilize the HuggingFace implementation of PEFT, which incorporates LoRA configurations to initialize LoRA-based fine-tuning of the Mistral model. By applying LoRA, we were able to reduce the number of training parameters from 3,837,112,320 to 85,041,152 (2.22% of the total), which are subsequently optimized using the AdamW optimizer.

A.5.3 MinMax Algorithm

Beyond solely relying on the Mistral model, we introduced an auxiliary model into the fine-tuning process following the implementation of the MinMax algorithm introduced by Korakakis and Vlachos (2023) to enhance the model's robustness in NLI training. This auxiliary model is designed to amplify the loss incurred in input spaces where the Mistral model encounters difficulties, effectively directing its focus towards areas of higher loss. The objective function for training is defined as:

$$J(\theta, \phi) = \min_{\theta} \max_{\phi} \frac{1}{n} \sum_{i=1}^{n} g_{\phi}(x_i, y_i) \cdot \mathcal{L}(f_{\theta}(x_i), y_i)$$

Here, θ denotes the mistral model parameters while ϕ denotes the auxiliary parameters that are optimized using standard optimization methods. $\mathcal{L}(f_{\theta}(x_i), y_i)$ is the cross entropy loss and $g_{\phi}(x_i, y_i)$ generates weights for each instance in the range (0,1).

A.5.4 Experiment Details

Here we provide the parameters used in our experiments for both the base and auxiliary models. For the base Mistral model, we used a LoRA configuration with the following parameters:

```
rank: 32
lora_alpha: 64
target_modules: [ q_proj, k_proj, v_proj,
o_proj, gate_proj, up_proj, down_proj,
lm_head ],
lora_dropout: 0.05
```

Parameters for fine-tuning mistral and auxiliary models are as follows:

```
Mistral:
    learning_rate: 3.3e-5
    batch_size: 4
    number_of_epoch: 1
    max_steps: 1000
Auxiliary:
    learning_rate: 5.8e-3
    hidden_size_1: 1024
    hidden_size_2: 64
```

Further system training and hyperparameter tuning details can be found at https://github.com/Bhuvanesh-Verma/RobustLLM

A.6 Results of Best Model with respect to Interventions and Sections

We examined the results on test data across various sections and interventions. Table 5 indicates that the `Adverse Event` section and *Numerical Contradiction* interventions yield the best performance.

	Type	No of Samples	F_1 Score		
			entailment	contradiction	macro avg
Section	Intervention	1542	0.58 (512)	0.75 (1030)	0.67
	Eligibility	1419	0.58 (485)	0.73 (934)	0.66
	Results	1235	0.58 (405)	0.80 (830)	0.69
	Adverse Events	1304	0.65 (439)	0.81 (865)	0.73
Intervention	Contradiction	1500	0 (0)	0.84 (1500)	0
	Numerical_contradiction	276	0 (0)	0.93 (276)	0
	Numerical_paraphrase	224	0.58 (91)	0.74 (133)	0.66
	Paraphrase	1500	0.73 (750)	0.70 (750)	0.72
	Text_appended	1500	0.57 (750)	0.70 (750)	0.63

Table 5: Intervention and Section-based results on test data using best model across both labels. Along with the F_1 score we also show number of instances.

Model	Token Length	Mode	Dev F_1
Mistral-7B-v0.1	1024	Remove	0.71
Mistral-7B-v0.1	1024	Truncate	0.72
Mistral-7B-v0.1	2048	Remove	0.72
Mistral-7B-v0.1	2048	Truncate	**0.73**

Table 6: Impact of different token length and strategy for handling long text

A.7 Handling Long Premise-Hypothesis Pairs

One of the challenges of processing CTRs is their extensive length when paired up to form a premise-hypothesis pair. The Mistral model allows for token lengths of up to 4096. We experimented with different token lengths to see how they impacted the model's performance. We trained models with token lengths of 1024 and 2048 and evaluated their performance on the dev set. From Table 6, we can see that increasing token length improved the results. We also tested the impact of truncating or removing text if it exceeded the token length. We observed that removing long text had a slight negative impact on the performance of the model. We used a token length of 4096 for our system development, with a truncation strategy in place for text that exceeds the token length limit.

A.8 Dataset Difficulty Analysis Details

For the MinMax weights approach, we first calculated the mean weight of correctly predicted instances. Every correctly predicted instance with a weight lower than the mean weight was selected as an easy instance (670). Similarly, for incorrectly predicted instances, we calculated their mean weight. Every incorrectly predicted instance with a weight higher than the mean weight was selected as a hard example (190).

With the data cartography strategy, we calculated the mean confidence for correctly predicted instances. Every instance with a confidence higher than the mean confidence was considered an easy-to-learn example (666). Similarly, every incorrectly predicted instance with a confidence lower than the mean confidence of incorrectly predicted instances was considered hard to learn (179).

Furthermore, we manually examined four examples, two from each method Minmax and data-cartography labeled as most hard or difficult to learn. Three out of the four examples target the Adverse Events section, with one targeting the Results section. Notably, all four examples involved numerical reasoning, suggesting that the model still struggles with numerical reasoning despite demonstrating promising results on numerical interventions in the test data. For more details, see Table 7.

A high overlap between the premise and hypothesis can lead to incorrect predictions of **entailment** relations, while low overlap can result in incorrect **contradiction** (Naik et al., 2018). Analysis of the hard examples in Figure 6 revealed that instances with high overlap predominantly belong to **contradiction** relations, however, were incorrectly predicted as **entailment** relations by the model. This phenomenon could be attributed to the model associating higher word overlap with **entailment** relations, as evidenced by the easy examples in Figure 6. However, such a correlation was not observed in the low word overlap region.

Using our trained model (*MINMAX-MEDNLI-FT-NLI4CT*), we generated explanations alongside

Difficulty	Type	Section				Label	
		Count	Intervention	Eligibility	Adverse Events	Results	contradiction	entailment
Easy	*single*	234	50	132	8	44	61	173
	comparison	88	44	37	1	6	29	59
Hard	*single*	59	3	3	36	17	50	9
	comparison	37	11	4	20	2	29	8
Easy-MinMax	*single*	433	75	225	21	112	99	334
	comparison	237	112	93	5	27	66	171
Hard-MinMax	*single*	102	9	9	60	24	83	19
	comparison	88	17	10	54	7	66	22
Easy-DataCartography	*single*	465	88	155	128	94	233	232
	comparison	201	95	45	41	20	132	69
Hard-DataCartography	*single*	108	13	27	39	29	76	32
	comparison	71	24	24	21	2	46	25

Table 7: Frequency of easy and hard examples across sections, instance type, and labels as identified by MinMax and data cartography methods. We also present combined results that is, the instances which are labeled easy and hard by both methods (Difficulty: Easy and Hard).

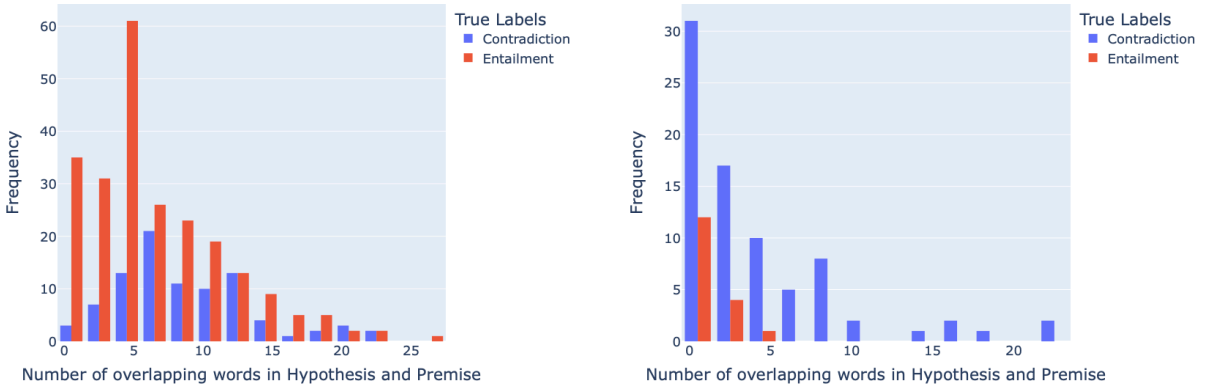


Figure 6: Word overlap between hypothesis and premise with respect to true labels in Hard examples (**on the right**) and Easy examples (**on the left**).

responses for each of these ten instances by increasing the number of generated tokens during the inference [3]. As outlined in the work of Swayamdipta et al. (2020), hard examples with low confidence scores may suggest mislabeled instances. We show two of these potential mislabeled instances in the Appendix A.9. Similarly, we also show the instances where the model confused cohorts and trials in Appendix A.10.

[3]This part was added after the first submission.

A.9 Potential Mislabeled Instances

<s>### Instruction: Read the input text and answer the following question with Yes or No.

Input:
Primary trial evidence are Outcome Measurement: Number of Participants With Objective Response Based on Data Review Committee's Assessment Number of participants with objective response based on assessment of confirmed complete response (CR) or confirmed partial response (PR) according to Response Evaluation Criteria in Solid Tumors version 1.0 (RECIST). CR is defined as disappearance of all target and non-target lesions. PR is defined as 30% decrease in sum of the longest dimensions (LDs) of the target lesions taking as reference the baseline sum LD according to RECIST. Confirmed responses are those that persist on repeat evaluation 4 weeks after initial documentation of response. Time frame: Day 1 of Cycle 2, every 6 weeks after Cycle 2, and at the end of Cycle 8.Results 1: Arm/Group Title: SUNITINIB+CAPECITABINE Arm/Group Description: Sunitinib was administered orally from Day 1 at the starting dose of 37.5 mg/day on a continuous daily dosing schedule in 21-day cycles. Capecitabine was administered orally from Days 1 to 14 every 21 days at a starting dose of 2,000 mg/m^2/day. Participants were monitored for toxicity, and sunitinib and/or capecitabine dosing could be interrupted or reduced according to individual tolerance. Participants with progressive disease (PD) or intolerable toxicity were considered for discontinuation from the study. Overall Number of Participants Analyzed: 63 Measure Type: Number Unit of Measure: participants Total Number of Participants with CR+PR: 19 Complete Response (CR): 0 Partial Response (PR): 19

Question: Does this imply that on assessment 0 the primary trial Participants had a confirmed disappearance of all target and non-target lesions.?

Response: No

Annotated Label: Entailment

<s>### Instruction: Read the input text and answer the following question with Yes or No.

Input:
Primary trial evidence are INTERVENTION 1: Moderated Group one 12-week online support group led by a professional healthcare providerINTERVENTION 2: Non-facilitated (Peer-led) 12-week online support in a peer-led format Secondary trial evidence are INTERVENTION 1: Sentinel Lymph Node Biopsy With Radiolabeled Methylene Blue One arm diagnostic using 1 mCi of 125-I Methylene blue dye to find sentinel lymph nodes

Question: Does this imply that Neither the primary trial or the secondary trial use Low Dose Magnesium Oxide, Biopsies or Mometasone in their intervention.?

Response: Yes

Annotated Label: Contradiction

A.10 Cohorts and Trial Confusion

Instruction: Read the input text and answer the following question with Yes or No.

Input:
Primary trial evidence are Adverse Events 1: Total: 4/42 (9.52%) Perforation, GI 1/42 (2.38%) Febrile neutropenia 1/42 (2.38%) Syncope 1/42 (2.38%) Rash/desquamation 1/42 (2.38%)

Question: Does this imply that 1/42 patients in cohort 2 of the primary trial fainted.?

Response: Yes

Annotated Label: Contradiction

Instruction: Read the input text and answer the following question with Yes or No.

Input:
Primary trial evidence are Adverse Events 1: Total: 267/744 (35.89%) Neutropenia *2/744 (0.27%) Anaemia *1/744 (0.13%) Leukopenia *1/744 (0.13%) Thrombocytopenia *1/744 (0.13%) Thrombotic thrombocytopenic purpura *1/744 (0.13%) Atrial flutter *1/744 (0.13%) Cardiac arrest *1/744 (0.13%) Myocardial ischaemia *1/744 (0.13%) Arrhythmia *0/744 (0.00%) Cardiac failure congestive *0/744 (0.00%)Adverse Events 2: Total: 67/736 (9.10%) Neutropenia *1/736 (0.14%) Anaemia *0/736 (0.00%) Leukopenia *0/736 (0.00%) Thrombocytopenia *0/736 (0.00%) Thrombotic thrombocytopenic purpura *0/736 (0.00%) Atrial flutter *0/736 (0.00%) Cardiac arrest *0/736 (0.00%) Myocardial ischaemia *0/736 (0.00%) Arrhythmia *2/736 (0.27%) Cardiac failure congestive *1/736 (0.14%) Secondary trial evidence are Adverse Events 1: Total: 6 Atrial fibrillation 1/67 (1.49%) Ventricular fibrillation 1/67 (1.49%) Gastrointestinal perforation 1/67 (1.49%) Periproctitis 1/67 (1.49%) General physical health deterioration 1/67 (1.49%) Escherichia sepsis 1/67 (1.49%) Pneumonia 1/67 (1.49%) Tumour pain 1/67 (1.49%) Renal failure acute 1/67 (1.49%) Pleurisy 1/67 (1.49%)

Question: Does this imply that The most common adverse events in the primary trial and the secondary trial is Neutropenia with a total of 3 cases across all cohorts.?

Response: No

Annotated Label: Entailment

UMUTeam at SemEval-2024 Task 8: Combining Transformers and Syntax Features for Machine-Generated Text Detection

Ronghao Pan[1], **José Antonio García-Díaz**[1],
Pedro José Vivancos-Vicente[2], **Rafael Valencia-García**[1]
[1] Facultad de Informática, Universidad de Murcia, Campus de Espinardo, 30100 Murcia, Spain
[2]VÓCALI Sistemas Inteligentes S.L., Parque Científico de Murcia,
Carretera de Madrid km 388. Complejo de Espinardo, 30100 Murcia, España
{ronghao.pan, joseantonio.garcia8, valencia}@um.es
pedro.vivancos@vocali.net

Abstract

These working notes describe the UMUTeam's participation in Task 8 of SemEval-2024 entitled "Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection". This shared task aims at identifying machine-generated text in order to mitigate its potential misuse. This shared task is divided into three subtasks: Subtask A, a binary classification task to determine whether a given full-text was written by a human or generated by a machine; Subtask B, a multi-class classification problem to determine, given a full-text, who generated it. It can be written by a human or generated by a specific language model; and Subtask C, mixed human-machine text recognition. We participated in Subtask B, using an approach based on fine-tuning a pre-trained model, such as RoBERTa, combined with syntactic features of the texts. Our system placed 23rd out of a total of 77 participants, with a score of 75.350%, outperforming the baseline.

1 Introduction

In the area of Natural Language Generation (NLG), advances such as GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020) and InstructGPT (Ouyang et al., 2022) have provided support for various writing tasks. The widespread adoption of Large Language Models (LLMs) such as ChatGPT and GPT-4 (Achiam et al., 2023) has led to an increase in machine-generated content across various platforms, including news, social media, education and science. While these models produce remarkably fluid responses, concerns have arisen about their potential to spread misinformation and disrupt established systems. Concerns remain about their misuse, particularly in scenarios such as academic dishonesty and scientific research, where AI-generated content may be presented as original work. The emergence of AI-generated scientific texts raises ethical and integrity concerns in academic publishing, requiring tools or models to distinguish between human-generated and AI-generated content (Ma et al., 2023).

Efforts to detect AI-generated text have primarily involved fine-tuning pre-trained models and developing detection systems. Recent studies have presented datasets and methods specifically designed for the detection of AI-generated scientific documents. However, challenges remain in achieving high performance and interpretability across different domains and models (Ma et al., 2023).

For this reason, Task 8 of SemEval (Wang et al., 2024), entitled "Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection", aims at identifying automatic systems for the detection of machine-generated text in order to mitigate its potential misuse. To this end, the task is divided into three subtasks that address two text generation paradigms: (1) full text, where a text is considered to be entirely written by a human or generated by a machine; and (2) mixed text, where a machine-generated text is refined by a human, or a text written by a human is paraphrased by a machine.

This shared task is divided into three subtasks:

- **Subtask A: Binary Human-Written vs. Machine-Generated Text Classification**. Determine whether a given full-text was authored by a human or generated by a machine. It offers two tracks: monolingual (English source only) and multilingual.
- **Subtask B: Multi-Way Machine-Generated Text Classification**. Given a full text, determine who generated it. It can be human-written or generated by a specific language model.
- **Subtask C. Human-Machine Mixed Text Detection**. Given a mixed text containing both human-generated and machine-generated

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 697–702
June 20-21, 2024

segments, identify the boundary where the transition from human-generated to machine-generated content occurs.

In this competition, the UMUTeam participated only in the **Subtask B** with an approach based on fine-tuning a pre-trained model such as RoBERTa combined with syntactic features of the text. Syntax features of the text refer to the writing style, such as token-level features (e.g. word length, part of speech, function word frequency and stop word ratio) and sentence-level features (e.g. sentence length).

During our experiments, we found that the syntactic features of texts can complement and improve the performance of pre-trained Transformer-based models and that RoBERTa is more suitable for this type of task.

The rest of this paper is organized as follows. First, Section 2 provides a summary of important details about the shared task setup. Second, Section 3 gives an overview of our system. Section 4 presents the specific details of our systems. Section 5 discusses the results of the experiments, and finally, the conclusions are presented in section 6.

2 Background

Recent advances in AI technology, particularly in the field of Natural Language Processing (NLP), have led to the emergence of many models capable of generating natural language using LLMs. These can produce remarkably fluent responses, and this has led to an increase in machine-generated content across multiple domains and platforms, including news, social media, education, and science.

LLMs face several technical and social challenges as they advance in NLP tasks. Recent research has shown that pre-trained LLMs can not only learn linguistic knowledge, but also reason about large amounts of acquired knowledge (Lewis et al., 2020). However, LLMs have other problems, such as hallucination, producing texts that contain information or details that are not based on reality or are completely invented; and asserting falsehoods as facts, which means that they can involuntarily produce texts that present false information as true.

The latest generative LLMs, such as GPT-3, are capable of producing highly fluent text, but they can produce inaccurate, toxic or unhelpful content. Some researchers have explored the use of reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) to adjust language models to better match user intent. ChatGPT, one of OpenAI's models based on GPT-3 and trained with RLHF, performs well in conversations with humans, demonstrating an understanding of user instructions and generating useful, reliable, honest and harmless text content.

Therefore, a growing number of studies have been conducted to analyze, recognize and identify text generated by AI, especially text generated by GPT. Current research focuses on two main areas: human behavior for recognizing text generated by AI and recognition models for identifying text generated by LLMs. For example, in (Guo et al., 2023), an approach was proposed to determine whether a text (in English and Chinese) was generated by ChatGPT or written by a human across different domains, while in (Shijaku and Canhasi, 2023), a model was developed to identify whether TOEFL essays were written by humans or generated by ChatGPT on a small dataset (126 essays for each).

There are other studies that focus on detecting fake information or fake news generated by LLMs. For example, in (Zellers et al., 2019), the Grover model was proposed to generate and detect examples of fake news. After the release of GPT-2, OpenAI proposed the GPT-2 generated text detector, which achieved a high F1 score. This detector was fine-tuned based on RoBERTa in a binary text classification format. In addition, many studies also use various data augmentation techniques to improve model performance in the classification task through external data that complements the model or simply increases the training set (Bayer et al., 2022). In paper (Ma et al., 2023), an approach was proposed to detect text generated by language models using different text features such as writing style, coherence, consistency, and argument logistics. The model with only syntax features (writing style) achieved the best result.

For this shared task, we used a fine-tuning approach of transformer-based models such as RoBERTa to create a detector for text generated by different LLMs. Unlike other existing studies on LLM-generated text detection, we have concatenated syntactic features during the fine-tuning process to improve its performance. The model evaluated for Subtask B is **RoBERTa** (Liu et al., 2019), a model based on Transformers, which was pre-trained on a large corpus of English data with Masked Language Model (MLM) goal. For this task, we evaluated the *base* version.

3 System overview

Figure 1 shows the architecture of our system. First, we extracted the syntactic features of the texts using the syntactic feature extractor and encoded the texts into a vector containing the dense representation of all the information contained in the text by the pre-trained models, i.e., the last hidden state of the model with text as input. Second, once the vector and syntactic features were obtained, we normalized the syntactic feature values and concatenated them with the text vector. Third, the fine-tuning process is performed, and a sequence classification layer is added on top of the pre-trained model. This layer takes the sequence representation generated by the pre-trained model and performs a classification based on the labels of the specific classification task. Finally, a performance evaluation is performed using the validation set.

3.1 Syntactic feature extractor

Syntactic linguistic features are those aspects related to the grammatical structure and organization of words in a sentence or paragraph (García-Díaz et al., 2022b). This can include elements such as sentence length, the frequency of certain parts of speech, the presence of function words, the number of stop words, etc. All of these features reflect the writing style that distinguishes different texts. In general, syntactic linguistic features have proven effective in NLP tasks such as author analysis (García-Díaz et al., 2022a) or hate speech identification (García-Díaz et al., 2023b).

The features used in this task are:

- **Average word length**. This is the average number of characters the words in the text have. It is calculated by adding the length of all words in the text and dividing that sum by the total number of words in the text.
- **POS tag frequency**. This is the frequency of Part of Speech (POS) grammatical tags in the text. Grammatical tags represent the grammatical categories of words in a text, such as nouns, verbs, adjectives, and so on. The frequency of POS tags indicates how often different grammatical categories occur in the text and can provide information about the structure and style of the text.
- **Average sentence length chars**. This is the average length of the sentences in the text, measured in characters. It is a measure of the complexity and readability of the text.
- **Average sentence length words**. This is the average length of the sentences in a text, measured in words. This metric shows the average number of words per sentence in the text.
- **Percentage of stopwords**. This is the percentage of stopwords in the text, relative to the total number of words in the text. Stopwords are common words that are often filtered out or eliminated during natural language processing because they occur so frequently and have little contextual meaning.
- **Punctuation Frequency**. Refers to the number of times that different punctuation marks, such as commas, periods, semicolons, etc., occur in a given text.
- **Special character Frequency**. Refers to the number of times characters other than letters and numbers occur in a given text. These special characters can include punctuation marks, mathematical symbols, control characters, emoticons, and other non-alphabetic symbols.

For the syntactic features, we used an open source tool called *authorstyle*[1], a package that allows to handle digital text forensics and stylometric corpora to extract stylometric features.

The embeddings of the texts refer to the numerical representation of the words or tokens in a high-dimensional vector space obtained by the tokenizers of the models. For this task, we normalized the syntactic feature values and concatenated them with the embeddings obtained by the tokenizers to perform RoBERTa fine-tuning to identify the author. It can be written by humans or generated by a specific language model.

4 Experimental setup

For Subtask B, we used the dataset provided by the organizers, which consists of two subsets: training and validation. Figure 2 shows the distribution of the training and validation sets. We can see that both the training and validation sets are balanced and that there are a total of 5 types of texts generated by different LLMs or by humans. The types

[1] https://github.com/mullerpeter/authorstyle

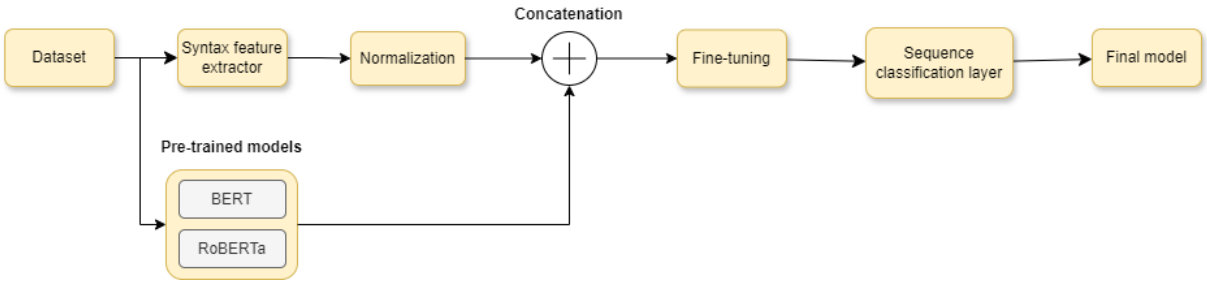


Figure 1: System architecture

are: davinci, bloomz, human, chatGPT, dolly and cohere.

We used the following fine-tuning hyperparameters: a batch size of 16 for both training and validation, 10 epochs, a learning rate of 2e-5, and a weight decay of 0.01.

During training, we used Macro-F1 as a reference. Macro-F1 is a measure used to evaluate the performance of a model in a multi-class classification problem. It calculates the average F1 score for each class individually, and then averages these scores to obtain an overall score. The macro F1 Score assigns equal weight to each class, regardless of its size or distribution in the data set. This means that all classes are equally important in the final scoring metric.

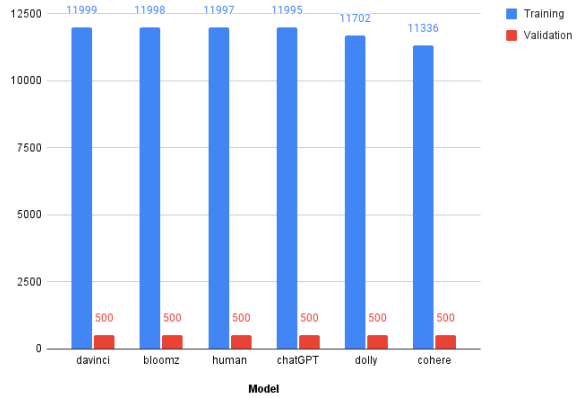


Figure 2: Training and validation set distribution of Subtask B.

5 Results

In the Table 1 we can see the official ranking of Subtask B. With a total of 77 contestants, we have achieved the twelfth-third best result, with an accuracy of 75.350, which is 0.744% higher than the baseline and 15.5% lower than the first.

In order to perform an error analysis and to observe the behavior of our model in predicting different classes of texts, we have generated the confusion matrix for our model based on the test set, as shown in Figure 3. Our analysis shows that our model has a strong predictive performance for texts generated by Bloomz, Dolly, ChatGPT and Davinci, reaching accuracies above 90%. However, when it comes to human-generated texts, our model shows a 27.73% tendency to misclassify them as generated by the Dolly model. In particular, when predicting texts generated by Cohere, our model tends to misclassify them as generated by Davinci at a rate of 70%, leading to a decrease in overall accuracy.

Table 1: Official results for the Subtask B.

Team	Rank	Accuracy
joeblack	1	90.850
tmarchitan	2	86.955
farawayxxc	3	84.328
halwhat	4	83.955
dianchi	5	83.478
	...	
UMUTeam	23	**75.350**
	...	
Baseline	-	74.606

6 Conclusion

This paper describes the participation of the UMUTeam in the 8th shared task of SemEval 2024, focused on the identification of automatic systems for the recognition of machine-generated text in order to mitigate its potential misuse. The task consisted of three subtasks: Subtask A, a binary classification task to determine whether a given full-text was written by a human or generated by a machine; Subtask B, a multi-class classification problem to determine, given a full-text, who gen-

Figure 3: The confusion matrix of our RoBERTa-based system in the test set.

erated it. It can be written by a human or generated by a specific language model; and Subtask C, mixed human-machine text recognition. In this shared task, we participated in Subtask B, using a approach based on fine-tuning a RoBERTa pretrained model with syntactic features of texts. In terms of results, our system achieved the 23rd position with a score of 75.350%, outperforming the baseline.

Due to our line of research, we will evaluate our system on texts containing figurative language (García-Díaz and Valencia-García, 2022) and financial language (García-Díaz et al., 2023a). On the one hand, the ambiguity and creativity of figurative language poses a challenge to the recognition of automatically generated text, as LLMs may have difficulty replicating the creative nuances of human-generated content. On the other hand, the recognition of automatically generated financial and business text is challenging due to specialized vocabulary and complex technical concepts. Ideally, LLMs must have deep domain-specific understanding to produce accurate content that requires regulatory compliance and accuracy, which requires careful review and validation against authoritative sources.

Acknowledgments

This work is part of the research projects LaTe4PoliticES (PID2022-138099OB-I00) funded by MICIU/AEI/10.13039/501100011033 and the European Regional Development Fund (ERDF)-a way to make Europe and LT-SWM (TED2021-131167B-I00) funded by MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. In addition, this work was funded by the Spanish Government, the Spanish Ministry of Economy and Digital Transformation through the Digital Transformation through the "Recovery, Transformation and Resilience Plan" and also funded by the European Union NextGenerationEU/PRTR through the research project 2021/C005/0015007. Mr. Ronghao Pan is supported by the "Programa Investigo" grant, funded by the Region of Murcia, the Spanish Ministry of Labour and Social Economy and the European Union - NextGenerationEU under the "Plan de Recuperación, Transformación y Resiliencia (PRTR)".

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Comput. Surv.*, 55(7).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

José Antonio García-Díaz, Ricardo Colomo-Palacios, and Rafael Valencia-García. 2022a. Psychographic traits identification based on political ideology: An author analysis study on spanish politicians' tweets posted in 2020. *Future Generation Computer Systems*, 130:59–74.

José Antonio García-Díaz, Francisco García-Sánchez, and Rafael Valencia-García. 2023a. Smart analysis of economics sentiment in spanish based on linguistic features and transformers. *IEEE Access*, 11:14211–14224.

José Antonio García-Díaz, Salud María Jiménez-Zafra, Miguel Angel García-Cumbreras, and Rafael Valencia-García. 2023b. Evaluating feature combination strategies for hate-speech detection in spanish using linguistic features and transformers. *Complex & Intelligent Systems*, 9(3):2893–2914.

José Antonio García-Díaz and Rafael Valencia-García. 2022. Compilation and evaluation of the spanish saticorpus 2021 for satire identification using linguistic features and transformers. *Complex & Intelligent Systems*, 8(2):1723–1736.

José Antonio García-Díaz, Pedro José Vivancos-Vicente, Angela Almela, and Rafael Valencia-García. 2022b. Umutextstats: A linguistic feature extraction tool for spanish. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6035–6044.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. 2023. Ai vs. human–differentiation analysis of scientific content generation. *arXiv*, 2301.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rexhep Shijaku and Ercan Canhasi. 2023. Chatgpt generated text detection. *Publisher: Unpublished*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

UMUTeam at SemEval-2024 Task 10: Discovering and Reasoning about Emotions in Conversation using Transformers

Ronghao Pan[1], **José Antonio García-Díaz**[1], **Diego Roldán**[2], **Rafael Valencia-García**[1]
[1] Facultad de Informática, Universidad de Murcia, Campus de Espinardo, 30100 Murcia, Spain
[2] DANTIA Tecnología S.L., Parque Empresarial de Jerez 10,
Calle de la Agricultura, 11407, Jerez de la Frontera, Cádiz, España
{ronghao.pan, joseantonio.garcia8, valencia}@um.es
droldan@dantia.es

Abstract

These notes describe the participation of the UMUTeam in EDiReF, the 10th shared task of SemEval 2024. The goal is to develop systems for detecting and inferring emotional changes in the conversation. The task was divided into three related subtasks: (i) Emotion Recognition in Conversation (ERC) in Hindi-English code-mixed conversations, (ii) Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations, and (iii) EFR in English conversations. We were involved in all three and our approach is based on a fine-tuning approach with different pre-trained models. After evaluation, we found BERT to be the best model for ERC and EFR and with this model we achieved the thirteenth best result with an F1 score of 43% in Subtask 1, the sixth best in Subtask 2 with an F1 score of 26% and the fifteenth best in Subtask 3 with an F1 score of 22%.

1 Introduction

Emotion, often defined as an individual's mental state associated with thoughts, feelings and behavior, has been categorized in various ways throughout history. Modern classifications include Plutchik's (Plutchik, 1982) eight primary types and Ekman's (Ekman, 1993) emphasis on facial expressions. In Natural Language Processing (NLP), emotion recognition has gained popularity for its applications in opinion mining, healthcare, etc. Although textual emotion recognition has been studied extensively, attention has recently shifted to Emotion Recognition in Conversation (ERC), driven by the availability of conversational data (Yeh et al., 2019) (Chen et al., 2018).

Conversation or dialogue is the main mode of information exchange between individuals, highlighting the prevalence of code-mixed (Kasper and Wagner, 2014), where multiple languages are integrated into the conversation. Despite extensive research on ERC, previous studies have largely focused on monolingual dialogues, neglecting code-mixed conversations. However, in the paper (Kumar et al., 2023a), the authors propose ERC models adapted to code-mixed dialogues, highlighting the need for datasets and resources in this area. Furthermore, they propose to incorporate common sense knowledge to better understand the emotions evoked in the conversation, and present a process to adapt existing English-based common sense knowledge graphs for code-mixed input.

ERC aims to identify emotions in sequences of utterances or dialogues rather than in isolated texts. In many cases, it is necessary to understand the emotional changes in a conversation is necessary in addition to identifying the speaker's emotion. However, understanding the emotional changes in a conversation is an challenging task that requires detailed analysis. Hence, the task of Emotion Flip Reasoning (EFR) (Kumar et al., 2022), which focuses on identifying the cause of a speaker's emotional change in a dialogue.

The EDiReF shared task (SemEval 2024) focuses on discovering and explaining the emotion change in the conversation (Kumar et al., 2024). It is divided into three subtasks: (1) **Subtask 1: ERC in Hindi-English code-mixed conversations**. Given a Hindi-English code-mixed dialog, the goal is to assign an emotion to each utterance from a predefined set of possible emotions (Kumar et al., 2023c); (2) **Subtask 2: EFR in Hindi-English code-mixed conversations**. Given a Hindi-English code-mixed dialog, the goal is to identify the trigger utterance(s) for an emotion flip in a multi-party conversation dialog (Kumar et al., 2022, 2023b); and (3) **Subtask 3: EFR in English conversations**. Given an English conversation, the goal is to identify the trigger utterance(s) for an emotion flip in a multi-party conversation dialog (Kumar et al., 2022, 2023b).

For this task, we propose an approach based on fine-tuning pre-trained Transformer-based models.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 703–709

In a nutshell, fine-tuning is a process by which a pre-trained model, previously trained on a specific task, is adjusted to adapt to a related but different task using a labeled dataset. In addition, a text processing process has been performed where, if possible, past and future conversations or emotions are added to the current user's sentence as input to the model. In this way, the model can have the context of the user's emotion in the past and future states.

These working notes are organized as follows. In Section 2, the reader will find a summary of important details about the task setup. Section 3 gives an overview of our system. Next, Section 4 presents the specific details of our systems. The results are then discussed and presented in Section 5. Finally, the conclusions are presented in Section 7.

2 Background

Sentiment Analysis (SA) is the study of human attitudes and feelings in specific situations, focusing on understanding emotions expressed through speech, voice, facial expressions and behavior. It typically identifies positive, negative and neutral emotions (Fu et al., 2023). In contrast, Emotion Recognition (ER) attempts to identify more nuanced emotions such as joy, hate and disgust, and modern classifications include Plutchik's (Plutchik, 1982) eight primary types and Ekman's (Ekman, 1993) emphasis on facial expressions. Emotion recognition spans text, audio and video modalities and differs from sentiment analysis in that it considers the context and interdependence between speakers within a conversation.

Multimodal emotion recognition has become an important research topic, mainly due to its potential applications in many challenging tasks such as dialog generation, user behavior understanding, multimodal interaction, and others. Therefore, a conversational emotion recognition system can be used to generate appropriate responses by analyzing the user's emotions. According to (Poria et al., 2019), ERC poses several challenges such as modeling the conversational context, emotion shifts of interlocutors, and others, which make the task more challenging. Recent works propose solutions based on multimodal memory networks (Hazarika et al., 2018). However, they are mostly limited to dyadic conversations and are therefore not scalable to ERC with multiple interlocutors. Furthermore, previous studies have largely focused on monolingual dialogues, neglecting code-mixed conversations (Kumar et al., 2023a).

In a conversation, utterances generally depend on the context of the conversation. This is also true for the emotions associated with them. In other words, the context acts as a set of parameters that can influence a person to make an utterance while expressing a certain emotion. This context can be modeled in different ways, for example using Recurrent Neural Networks (RNN) and Memory Networks (Hazarika et al., 2018) (Serban et al., 2017). Public datasets available for multimodal emotion recognition in conversation, such as IEMOCAP (Busso et al., 2008) and SEMAINE (McKeown et al., 2010), have facilitated a significant number of research projects, but they also have limitations due to their relatively small number of total utterances and the lack of multipart conversations.

Understanding the emotional flips in a conversation requires a detailed analysis. This is where Emotional Flip Reasoning (EFR) comes in, which focuses on identifying the cause of a speaker's emotional flip in a dialogue. The EFR process consists of three stages (Kumar et al., 2022): identifying the utterance in which the emotional flip occurs, identifying the triggers responsible for the change, and assigning psychologically motivated instigator labels to the triggers to explain the emotional flip. Therefore, the EFR task has the potential to improve the user experience in a conversational dialog system, especially in the generation of empathetic responses (Lin et al., 2019), (Ma et al., 2020).

In recent years, with the rapid development in the field of NLP, many pre-trained models based on Transformer have emerged. These models are trained on large corpora of unlabeled text and, due to their transfer learning capability, can be adapted to different tasks such as classification, translation, response generation without the need of a large training corpus. For example, (García-Díaz et al., 2023) and (García-Díaz and Valencia-García, 2022) demonstrated the effectiveness of Transformers-based models for identifying hate speech and satire. Therefore, in this study, different pre-trained models were evaluated for the ERC and EFR tasks.

The models evaluated are: (1) XLM-RoBERTa-base (Conneau et al., 2019); (2) DeBERTa-V3-base(He et al., 2021); and (3) BERT (Devlin et al., 2018). For the ERC and EFR tasks, we evaluated the basic version and the version without the mask,

which removes the accent markers.

3 System overview

Figure 1 shows the general architecture of our approach for the three subtasks, which is mainly divided into two modules: data processing and fine tuning.

In the processing module, for Subtask 1 (ERC), we first translated the statements into English, since most language models are pre-trained in English and have shown good performance in the emotion identification and sentiment analysis tasks. They were then grouped by user, as this provides a coherent context for analyzing their emotional state, rather than adding conversational context from other speakers. Therefore, by examining all the interventions made by the same speaker, we gain a deeper understanding of their emotional state at the time of the target intervention. Furthermore, we believe that adding more context could introduce noise and reduce the performance of the models. Once grouped, for each current statement of the user, the previous statement was concatenated with the next by a semicolon. For example, for statement U3 from a particular user, the input to the model would be `U2;U3;U4`. For subtasks 2 and 3 (EFR) in addition to concatenating the previous and subsequent statements from the same user for each current statement, the emotion of each statement is added. For example, for statement U3, the input to the model would be `U2-e2;U3-e3;U4-e4`, where *e* represents the user's emotion at that moment. The figure 2 shows examples of processing for the user *Ross* in a specific conversation.

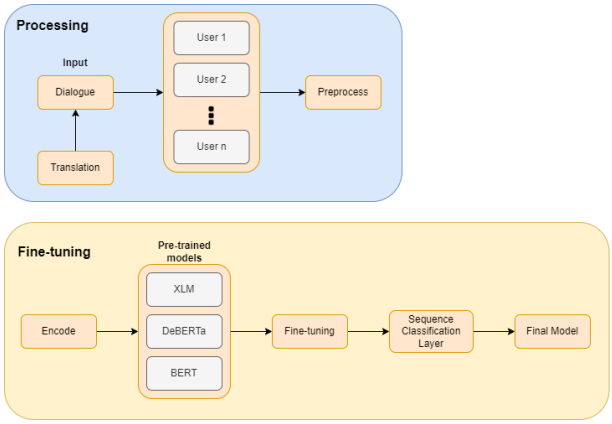


Figure 1: System architecture

In the fine tuning module (see Figure 1), the inputs are first tokenized according to the tokenizers of the pre-trained models. Next, the pre-trained model is loaded as the basis for the classification task. Next, a sequence classification layer is added on top of the pre-trained model. This layer takes the last hidden state generated by the pre-trained model and performs classification based on the labels of the specific classification task. In this case, we used the sequence classification layer from the *Transformers*[1] library for each pre-trained model. Finally, the tuning is performed out and a performance is evaluated using the validation set.

4 Experimental setup

To train the three subtasks, we used the data set provided by the organizers, which consists of a training set and a validation set. In Figure 3 and Table 1 we can see the distribution of the training and validation sets for the three subtasks.

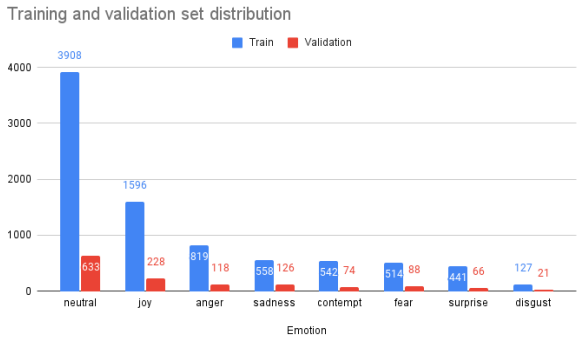


Figure 3: Training and validation set distribution of Subtask 1.

Table 1: Training and validation set distribution of Subtask 2 and 3.

Set	Triggers	No triggers
Subtask 2		
Train	6542	92235
Validation	434	7028
Subtask 3		
Train	5575	29416
Validation	494	3027

For all three subtasks (1, 2, 3), we used the same fine-tuning hyperparameters, namely: a batch size of 8 for both training and validation, 10 epochs, a learning rate of 2e-5, and a weight decay of 0.01. During training, we used the weighted F1 as a reference. To evaluate the three subtasks, the organizers

[1] https://github.com/huggingface/transformers

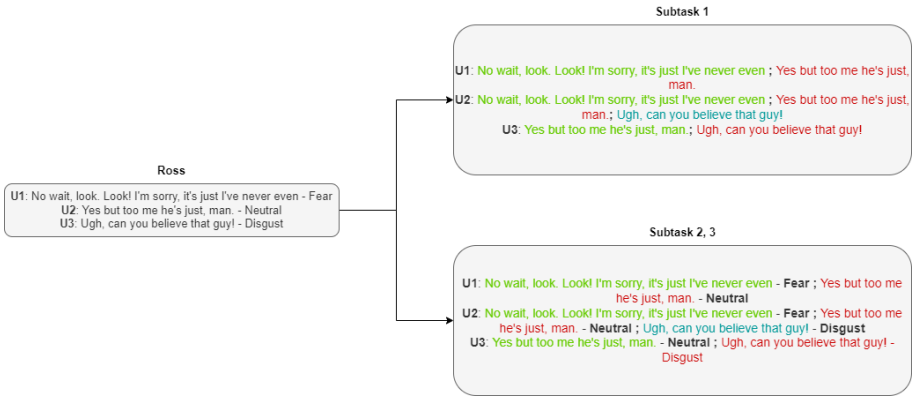


Figure 2: Examples of processing for subtasks 1, 2, and 3.

used the weighted F1, an evaluation metric used in classification problems that takes into account the class imbalance in the data. While the traditional F1 score calculates the harmonic mean of precision and recall for all classes equally, the weighted F1 score weights these measures according to the number of samples in each class.

5 Results

Table 2 shows the results obtained on the test set with different models for Subtask 1 on the ERC. We can see that the XLM-R model obtained the best result with a weighted F1 score of 42.878%, followed by BERT with a weighted F1 score of 42.691%.

Table 2: Evaluation of different pre-trained models in test set of Subtask 1.

Model	W-R	W-P	W-F1
Subtask 1			
XLM-R	**44.9367**	42.1941	**42.878**
DeBERTa	43.5443	41.0664	41.7686
BERT	44.8734	**42.4540**	42.6910

Table 3 shows the results of Subtask 2, which is an EFR task, on a dataset of Hindi-English code-mixed conversations. The evaluation metric is the F1 score of the triggers, and it can be seen that BERT is the only model that obtained a score greater than 0, with 25.8721% in F1 score. The XLM-R and DeBERTa models were not able to predict emotion change triggers well because they were fine-tuned with the same hyperparameters, so it may be necessary to use different hyperparameters, such as a smaller learning rate. Therefore, as a future line, it is proposed to perform hyperparameter tuning to fine-tune the models to achieve better performance.

Regarding Subtask 3, which has the same objective as Subtask 2, but on a dataset of English code-mixed conversations, it can be observed that BERT and DeBERTa are the only two models that have obtained an F1 score greater than 0, with 22.4764% for BERT and 17.1111% for DeBERTa (see Table 3).

Table 3: Evaluation of different pre-trained models in test set of Subtask 2 and 3.

Model	Recall	Precision	F1
Subtask 2			
XLM-R	0.0	0.0	0.0
DeBERTa	0.0	0.0	0.0
BERT	**21.3942**	**32.7206**	**25.8721**
Subtask 3			
XLM-R	0.0	0.0	0.0
DeBERTa	13.1737	24.4057	17.1111
BERT	**19.3328**	**87.9103**	**22.4764**

Therefore, we have chosen the BERT model for this task, since it outperforms the other models in all three subtasks, except for the first, where it is 0.187% worse than XLM-RoBERTa, which does not exceed 1%. In this case, we have obtained the thirteenth position in Subtask 1, the sixth in Subtask 2 and the fifteenth in Subtask 3.

6 Error analysis

For error analysis, we extracted the confusion matrix from BERT using the Subtask 1's test sets. A confusion matrix is a tool used in error analysis, especially in classification scenarios, by illustrating

the performance of a model in predicting true class labels compared to the model-predicted classes.

In Figure 4, we can see that our system tends to confuse the *Neutral* emotion in the ERC task, due to the unbalanced training set provided by the organizers, where the *Neutral* emotion occupies the highest percentage. Furthermore, the disgust emotion was not correctly identified in any case.

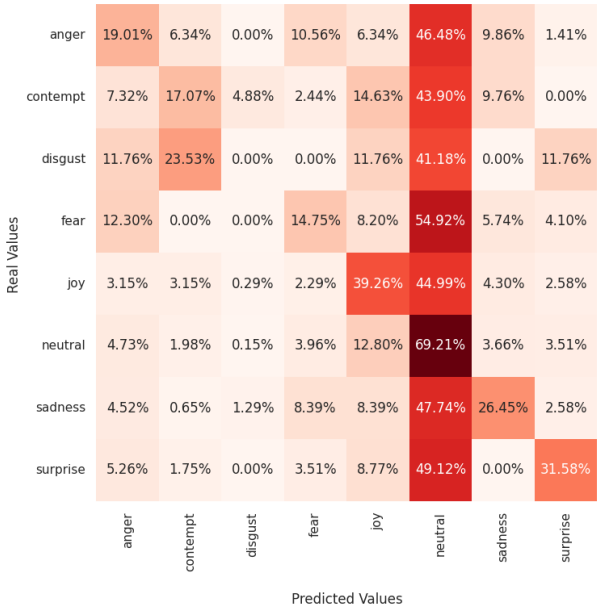


Figure 4: BERT confusion matrix in the test set of subtask 1.

Table 4 shows a classification report of our model in the EFR task of Hindi-English code-mixed conversation (Subtask 2). We can see that our system tends to identify instances as "No triggers" and has a higher recall due to the imbalance in the training set, which contains more instances of "no triggers". As for Subtask 3, the same phenomenon occurs as in Subtask 2, as shown in Table 5.

Table 4: BERT's classification report of Subtask 2 in the test set.

	Precision	**Recall**	**F1**
No triggers	95.5918	97.4842	96.5287
Triggers	32.7206	21.3942	25.8721
Macro avg	64.1562	59.4392	61.2004
Weighted avg	92.1907	93.3680	92.7065

Table 5: BERT's classification report of Subtask 3 in the test set.

	Precision	**Recall**	**F1**
No triggers	87.9103	91.7570	89.7924
Triggers	26.8409	19.3328	22.4764
Macro avg	57.3756	55.5449	56.1344
Weighted avg	79.6494	81.9602	80.6866

7 Conclusion

We have described the UMUTeam's participation in the 10th shared task 10 of SemEval 2024, the goal of which was to develop models for detecting and reasoning about the emotion change in the conversation. The task consists of three subtasks: (i) Emotion Recognition in Conversation (ERC) in Hindi-English code-mixed conversations, (ii) Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations, and (iii) EFR in English conversations.

For all three subtasks, we used the fine-tuning approach of pre-trained models and performed a text processing process where, where possible, previous and future conversations or emotions are added to the current user's sentence as input to the model. In terms of results, our system achieved the thirteenth best result in Subtask 1 with an F1 of 43%, the sixth best in Subtask 2 with an F1 of 26%, and the fifteenth best in Subtask 3 with an F1 of 22%.

The study of emotional shifts provides a valuable insights for understanding psychographic characteristics in author profiling in the political context. Political communication is inherently intertwined with emotional appeals, and the ability to identify patterns of emotional shifts provides insight into the psychological makeup of political authors. Therefore, we plan to further validate the effectiveness of emotion flip inference by applying it to our PoliticES 2022 and 2023 datasets (García-Díaz et al., 2022; Garcia-Díaz et al., 2023) thus, contributing to a more comprehensive understanding of the ideologies, motivations, and communication strategies of political figures.

Acknowledgments

This work is part of the research projects LaTe4PoliticES (PID2022-138099OB-I00) funded by MICIU/AEI/10.13039/501100011033 and the European Regional Development Fund (ERDF)-a way to make Europe and

LT-SWM (TED2021-131167B-I00) funded by MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. In addition, this work was funded by the Spanish Government, the Spanish Ministry of Economy and Digital Transformation through the Digital Transformation through the "Recovery, Transformation and Resilience Plan" and also funded by the European Union NextGenerationEU/PRTR through the research project 2021/C005/00149877. Mr. Ronghao Pan is supported by the "Programa Investigo" grant, funded by the Region of Murcia, the Spanish Ministry of Labour and Social Economy and the European Union - NextGenerationEU under the "Plan de Recuperación, Transformación y Resiliencia (PRTR)".

References

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Lun-Wei Ku, et al. 2018. Emotionlines: An emotion corpus of multi-party conversations. *arXiv preprint arXiv:1802.08379*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Paul Ekman. 1993. Facial expression and emotion. *American psychologist*, 48(4):384.

Yao Fu, Shaoyang Yuan, Chi Zhang, and Juan Cao. 2023. Emotion recognition in conversations: A survey focusing on context, speaker dependencies, and fusion methods. *Electronics*, 12(22).

José Antonio García-Díaz, Salud María Jiménez-Zafra, Miguel Angel García-Cumbreras, and Rafael Valencia-García. 2023. Evaluating feature combination strategies for hate-speech detection in spanish using linguistic features and transformers. *Complex & Intelligent Systems*, 9(3):2893–2914.

José Antonio Garcia-Díaz, Salud María Jiménez-Zafra, María-Teresa Martín-Valdivia, Francisco García-Sánchez, Luis Alfonso Ureña-López, and Rafael Valencia-García. 2023. Overview of politices at iberlef 2023: Political ideology detection in spanish texts. *Procesamiento del Lenguaje Natural*, 71:409–416.

José Antonio García-Díaz, Salud María Jiménez-Zafra, María-Teresa Martín Valdivia, Francisco García-Sánchez, L Alfonso Ureña-López, and Rafael Valencia-García. 2022. Overview of politices 2022: Spanish author profiling for political ideology. *Procesamiento del Lenguaje Natural*, 69:265–272.

José Antonio García-Díaz and Rafael Valencia-García. 2022. Compilation and evaluation of the spanish saticorpus 2021 for satire identification using linguistic features and transformers. *Complex & Intelligent Systems*, 8(2):1723–1736.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2018, page 2122. NIH Public Access.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Gabriele Kasper and Johannes Wagner. 2014. Conversation analysis in applied linguistics. *Annual Review of Applied Linguistics*, 34:171–212.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Md Shad Akhtar, Tanmoy Chakraborty, et al. 2023a. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. *arXiv preprint arXiv:2310.13080*.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023b. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023c. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Zhaojiang Lin, Andrea Madotto, Jamin Shin, Peng Xu, and Pascale Fung. 2019. Moel: Mixture of empathetic listeners. *arXiv preprint arXiv:1908.07687*.

Yukun Ma, Khanh Linh Nguyen, Frank Z Xing, and Erik Cambria. 2020. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70.

Gary McKeown, Michel F. Valstar, Roderick Cowie, and Maja Pantic. 2010. The semaine corpus of emotionally coloured character interactions. In *2010 IEEE International Conference on Multimedia and Expo*, pages 1079–1084.

Robert Plutchik. 1982. A psychoevolutionary theory of emotions.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953.

Iulian Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the AAAI conference on artificial intelligence*, 1.

Sung-Lin Yeh, Yun-Shao Lin, and Chi-Chun Lee. 2019. An interaction-aware attention network for speech emotion recognition in spoken dialogs. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6685–6689. IEEE.

TM-TREK at SemEval-2024 Task 8: Towards LLM-Based Automatic Boundary Detection for Human-Machine Mixed Text

Xiaoyan Qu and **Xiangfeng Meng**
Samsung R&D Institute China-Beijing
{xiaoyan11.qu, xf.meng}@samsung.com

Abstract

With the increasing prevalence of text generated by large language models (LLMs), there is a growing concern about distinguishing between LLM-generated and human-written texts in order to prevent the misuse of LLMs, such as the dissemination of misleading information and academic dishonesty. Previous research has primarily focused on classifying text as either entirely human-written or LLM-generated, neglecting the detection of mixed texts that contain both types of content. This paper explores LLMs' ability to identify boundaries in human-written and machine-generated mixed texts. We approach this task by transforming it into a token classification problem and regard the label turning point as the boundary. Notably, our ensemble model of LLMs achieved first place in the 'Human-Machine Mixed Text Detection' sub-task of the SemEval'24 Competition Task 8. Additionally, we investigate factors that influence the capability of LLMs in detecting boundaries within mixed texts, including the incorporation of extra layers on top of LLMs, combination of segmentation loss, and the impact of pretraining. Our findings aim to provide valuable insights for future research in this area.

1 Introduction

Large language models (LLMs), particularly since the debut of ChatGPT, have made significant advancement and demonstrated the ability to produce coherent and natural-sounding text across a wide range of applications. However, the proliferation of generated text has raised concerns regarding the potential for misuse of these LLMs. One major issue is the tendency of LLMs to produce hallucinated content, resulting in text that is factually inaccurate, misleading, or nonsensical. Inappropriate utilization of LLMs for text generation purposes, such as in news articles (Zellers et al., 2019), social media posts (Fagni et al., 2021), and app reviews (Martens and Maalej, 2019), can propagate misinformation and influence public perceptions. Furthermore, the use of machine-generated text can also facilitate academic dishonesty. Therefore, accurately distinguishing between human-authored and machine-generated texts is crucial in order to address these challenges effectively.

The majority of existing studies addressing this challenge have approached it as a machine-generated text classification problem, aiming to determine whether a given text is generated by LLMs or not. However, this approach assumes that the text is either completely machine-generated or entirely human-written. With the increasing collaboration between humans and AI systems, mixed texts containing both human-authored and machine-generated portions have emerged as a new scenario that simple machine-generated text classification methods cannot effectively address (Dugan et al., 2023). Therefore, a more nuanced approach to machine-generated text classification for mixed texts is necessary.

This study addresses the challenge of token-level boundary detection in mixed texts, where the text sequence starts with a human-written segment followed by a machine-generated portion. The objective is to accurately determine the transition point between the human-written and LLM-generated sections. To achieve this, we frame the task as a token classification problem, thus the turning point of the label sequence will be the boundary. Through experiments utilizing LLMs that excel in capturing long-range dependencies, we demonstrate the effectiveness of our approach. Notably, by leveraging an ensemble of multiple LLMs to harness the robust of the model, we achieved first place in Task 8 of SemEval'24 competition.

Furthermore, we explore factors that impact the effectiveness of LLMs in boundary detection, including the integration of additional layers on top of LLMs, the combination of segmentation loss and pretraining techniques. Our experiments indicate

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 710–715

that optimizing these factors can lead to significant enhancements in boundary detection performance.

The main contribution of this papers include:

1) We explore LLMs' capability to detect boundaries within human-machine mixed texts, compare the performance of various LLMs, and present a benchmark based on the new released data set. And we rank 1st in the corresponding SemEval'24 competition (Wang et al., 2024).

2) We examine factors that impact boundary detection in mixed texts, including additional layers on top of LLMs, introduce of segment loss functions, and pretraining technique. We aim to provide valuable insights for future research in this field.

2 Related Work

Previous research has predominantly focused on machine-generated text classification (Crothers et al., 2023; Jawahar et al., 2020), where the text is attributed to either human authors or large language models. The objective is to determine whether a given text is human-written or specifically generated by a particular LLM. These studies can be classified into two main categories: metric-based methods and model-based methods. Metric-based approaches leverage metrics such as word rank, predicted distribution entropy, and log-likelihood (Mitchell et al., 2023; Gehrmann et al., 2019; Venkatraman et al., 2023). On the other hand, model-based methods involve training models on labeled data (Liu et al., 2022). However, these methods are not directly applicable to boundary detection for mixed human-machine texts.

Recently, there are a few works investigating the detection of mixed human-machine text. These texts consist of both human-written and machine-generated content, and the objective is to accurately identify the boundary between these two segments. Dugan et al. (2023) delved into the human ability to discern boundaries between human-written and machine-generated text. Their study revealed significant variations in annotator proficiency and analyzed the impact of various factors on human detection performance. Zeng et al. (2023) were the first to formalize the task as identifying transition points between human-written and AI-generated content within hybrid texts, and they examined automated approaches for boundary detection.

One limitation of these studies is that the transitions typically occur between sentences rather than at the word level. This paper aims to address the token-level boundary detection of mixed texts.

3 Methodology

3.1 Task Formulation

The task is presented as a sub-task 'Subtask C: Human-Machine Mixed Text Detection' in SemEval'24 Task 8[1]. The task is defined as follows: for a hybrid text $< w_1, w_2, ..., w_n >$ with a length of n that includes both human–written and machine-generated segments, the objective is to determine the index k, at which the initial top k words are authored by humans, while the subsequent are generated by LLMs. The evaluation metric for this task is Mean Absolute Error (MAE). It measures the absolute distance between the predicted word and the actual word where the switch between human and machine occurs.

We transform the task of boundary detection in mixed texts into a token classification task, aligning it with the competition baselines. Token classification involves assigning a label to each token within a text sequence. In boundary detection tasks, we utilize two labels to indicate whether each token was written by humans or generated by LLMs. By predicting the label of each token in the text sequence, we can identify the specific word that signifies the boundary between the human-written and machine-generated portions of the text.

3.2 LLM based Boundary Detection

We explores LLMs' capability to detect boundaries for human-machine mixed texts. The framework of this paper is shown in Figure 1. Given the labeled dataset containing boundary indices, we first map these indices to assign each token a label denoting whether it originates from human writing or LLM generation. Subsequently, we harness the capabilities of LLMs by fine-tuning them for the task of classifying each token's label. To enhance performance further, we employ an ensemble strategy that consolidates predictions from multiple fine-tuned models. Additionally, we explore various factors that impact the effectiveness of LLMs in boundary detection.

3.2.1 LLMs Supporting Long-range Dependencies

Our objective is to facilitate boundary detection in long text sequences, thereby necessitating the

[1] https://github.com/mbzuai-nlp/SemEval2024-task8/tree/main

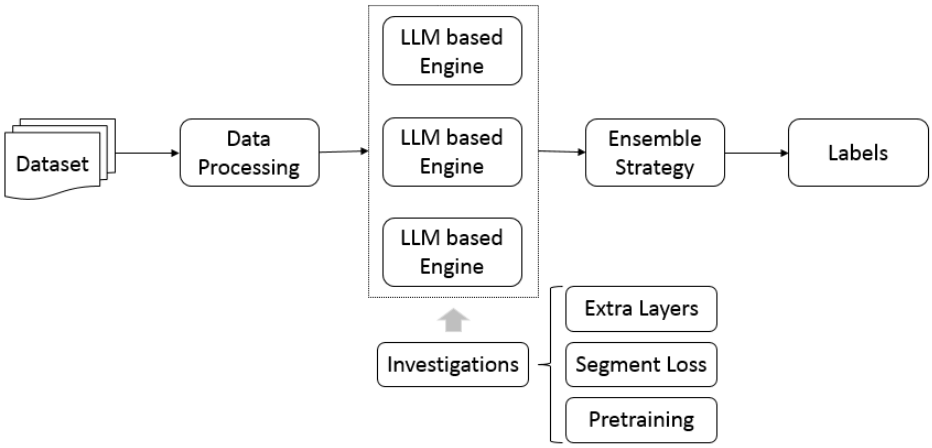


Figure 1: Framework of this paper

utilization of LLMs capable of handling long-range dependencies. Within this study, we investigate the performance of Longformer, XLNet, and BigBird models on boundary detection.

Longformer (Beltagy et al., 2020) utilizes a combination of global attention and local window-based attention mechanisms, which enables the model's capability to capture both short-range and long-range dependencies effectively.

XLNet (Yang et al., 2019) introduces a new training objective called permutation language modeling, and considers all possible permutations of the input tokens during training. This allows XLNet to capture bidirectional context more effectively and mitigate the limitations of autoregressive models.

BigBird (Zaheer et al., 2020) introduces a novel sparse attention mechanism that allows the model to scale to longer sequences while maintaining computational efficiency.

3.2.2 Exploration of Potential Factors

Except for the direct usage of LLMs, we investigate factors that may influence the capability of LLMs in boundary detection task.

Extra Layers on top of LLMs While LLMs demonstrate remarkable proficiency in comprehending semantics and generating coherent text, the addition of supplementary layers on top of LLMs has the potential to yield further improvements for downstream tasks. Therefore, we evaluate the impact of additional layers, such as LSTM and CRF, when integrated with LLMs, to ascertain their potential contributions to enhancing performance in boundary detection tasks.

Segment Loss Function Token classification involves assigning specific categories to individual tokens within a text sequence based on their semantic content. Typically, evaluation metrics gauge the average accuracy of category assignments across all tokens. However, in the context of boundary detection, the labels of tokens situated at or in proximity to the boundary hold greater significance. To bridge this gap, we introduce loss functions capable of assessing segment accuracy for both the human-written and machine-generated segments, such as the dice loss function. These loss functions, commonly utilized in image segmentation tasks that entail dividing an image into distinct segments, are anticipated to enhance performance in boundary detection tasks.

Pretrain and Fine tune Within the SemEval'24 competition, a total of 4,154 cases are presented for this task. The remaining sub-tasks revolve around human-machine text classification and aim to classify a given text into either human-generated or LLM-generated. A natural idea is to initiate pretraining utilizing the text classification data, followed by fine-tuning on the bounary detection data to enhance the model's overall generalization capability.

Two distinct pretraining approaches are employed. In the Pretrain 1, human-written texts and machine-generated texts are concatenated to form a new boundary detection dataset in sentence level. Within this novel dataset, boundaries are identified at the juncture where human-written and machine-generated sentences merge. Models are trained initially on the sentence-level dataset and subsequently fine-tuned using the 4,154 cases provided.

	Training data	Dev data
Number	3649	505
Average Length	263	230
Max Length	1397	773
Average Index	71	68

Table 1: Statistics of the boundary detection data set

	Train	Dev
binary text classification	119,757	5,000
multi-way text classification	71,027	3,000

Table 2: Statistics of two other data sets for human-machine text classification tasks

Method	MAE
Longformer (baseline)	4.11
Longformer-large	3.58
XLNet-large	**2.44**
BigBird-large	5.91

Table 3: Performance of varied LLMs

Method	MAE
XLNet-large	2.44
XLNet-large vote [2]	2.22

Table 4: Performance of multiple XLNet ensembles

In the Pretrain 2, a binary text classification model incorporating both an LLM and a linear layer atop the LLM is initially trained. Subsequently, the weights of the LLM are utilized for fine-tuning in the boundary detection task.

4 Experiments

4.1 Dataset

The data is an extension of the M4 dataset (Wang et al., 2023). It consists of 3,649 train cases and 505 dev cases, each with text content and gold boundary index. The boundary index denotes the position of the word split caused by a change, with white space serving as the delimiter. Data statistics are shown in Table 1.

In addition, SemEval'24 Competition also presented two datasets related to binary and multi-way text classification tasks. We investigate whether these additional datasets could potentially enhance boundary detection performance by pretraining techniques. The details of the two supplementary datasets are shown in Table 2.

4.2 Experimental Results

4.2.1 Performance of different LLMs

We investigate three LLMs renowned for their ability to handle long-range dependencies: Longformer, XLNet, and BigBird. We exclusively employ the large versions of these models. Moreover, as a benchmark for the competition, we utilize Longformer-base. The performance metrics of these four models are outlined in Table 3.

As depicted in Table 3, we observe that Longformer-large outperforms Longformer-base, owing to its increased parameter count. Among the four algorithms, XLNet achieves the best performance, with an MAE of 2.44. This represents a reduction of 31.84% compared to Longformer-large and a substantial 58.71% decrease compared to BigBird-large. One potential explanation is that the consideration of all possible permutations of input tokens during training help XLNet capture bidirectional context more effectively.

The winning approach in SemEval'24 Competition is founded on ensembles of 2 XLNet with varied seeds. It involves a simple voting process of the output logits from the diverse XLNet models. As shown in Table 4, the voting strategy results in a decrease in MAE from 2.44 to 2.22.

4.2.2 Performance of LLMs with extra layers

We select Longformer-large as our baseline model and examined the impact of incorporating extra LSTM, BiLSTM, and CRF layers(Huang et al., 2015) on boundary detection. The experimental results are detailed in Table 5. Integration of LSTM and BiLSTM layers with Longformer leads to significant improvements, with a decrease in MAE by 10.61% and 23.74%, respectively. Conversely, the addition of a CRF layer to Longformer-large yields unsatisfactory results. One plausible explanation could be the lack of clear dependencies between the two labels (0 and 1) in the token classification task, unlike in tasks such as named entity extraction.

4.2.3 Performance with segment loss functions

We investigate the impact of employing segment loss functions commonly utilized in image segmentation (Jadon, 2020) on boundary detection. The selected loss functions consist of BCE dice loss, Jaccard loss, Focal loss, Combo loss and Tversky loss. Additionally, we introduce a novel loss function BCE-MAE by simply adding BCE and MAE.

[2]The approach that ranks 1st in sub-task C leaderboard

Method	MAE
Longformer-large	3.58
Longformer-large + LSTM	3.20
Longformer-large + BiLSTM	2.73
Longformer-large + CRF	5.86

Table 5: Performance of adding extra layers

Method	MAE	Method	MAE
base (BCE loss)	3.58	Dice loss	3.80
BCE-dice loss	**3.14**	Jaccard loss	3.60
Focal loss	3.40	Combo loss	**3.09**
Tversky loss	3.69	BCE-MAE loss	**2.99**

Table 6: Performance of different segment loss functions

We utilize Longformer-large as the baseline, which by default adopts the binary cross-entropy loss. We explore the impact of adjusting the loss functions and present the results in Table 6. Among these variations, BCE-dice loss, Combo loss, and the BCE-MAE loss demonstrate superior performance compared to the default BCE loss.

Both BCE-dice loss and Combo loss are initially designed to integrate binary cross-entropy and Dice loss using different weighting schemes to enhance the performance of binary image segmentation. Dice loss serves as a metric for assessing the overlap between the predicted segmentation and the ground truth mask. The introduction of the Dice loss enables a balance between segmentation accuracy and token-wise classification accuracy, resulting in anticipated performance improvements. Compared to the benchmark, the MAE decreases by 12.29% and 13.69%, respectively.

The BCE-MAE loss incorporates MAE loss during the training stage, aligning with the evaluation metric used in the competition. As anticipated, the MAE metric decreases by 16.48%.

4.2.4 Performance of LLMs with pretraining

For both two pretraining approaches, we employ three different settings: directly utilizing Longformer-large for pretraining and fine-tuning; and incorporating additional LSTM and BiLSTM layers, respectively. Table 7 presents the results.

In Pretrain 1, the datasets from the other two subtasks are concatenated to create a new dataset in which the segmentation is on a sentence level, similar to previous studies. Sentence-level boundary detection is akin to token-level boundary detection. So the pretraining of sentence-level data is anticipated to obtain extra gains. The table indicates

Pretraining	Method	MAE
No pretrain	Longformer-large	3.58
Pretrain 1	Longformer-large	3.26
	+ LSTM	2.85
	+ BiLSTM	2.84
Pretrain 2	Longformer-large	68.52
	+ LSTM	3.04
	+ BiLSTM	2.72

Table 7: Performance of Longformer-large with pretraining

that simply employing Longformer-large with pretraining reduces the MAE from 3.58 to 3.26. By incorporating LSTM and BiLSTM layers, the MAE is further reduced to 2.85 and 2.84, respectively.

In Pretrain 2, we initially utilize Longformer to classify whether a given text is human-written or machine-generated. When the pretrained Longformer is fine-tuned directly on boundary detection, only inserting a new linear layer yields poor performance. However, with the inclusion of additional LSTM and BiLSTM layers, it can achieve comparable performance to that of Pretrain 1, reaching 3.04 and 2.72, respectively.

The results of the two pretraining approaches indicate that pretraining on either sentence-level boundary detection or the binary human-machine text classification task can enhance LLMs' capability to detect token-wise boundaries in mixed texts.

5 Conclusion

This paper introduces LLM-based methodology for detecting token-wise boundaries in human-machine mixed texts. Through an investigation into the utilization of LLMs for boundary detection, we have achieved optimal performance by leveraging an ensemble of XLNet models in the SemEval'24 competition. Furthermore, we explore factors that could affect the boundary detection capabilities of LLMs. Our findings indicate that (1) loss functions considering segmentation intersection can effectively handle tokens surrounding boundaries; (2) supplemental layers like LSTM and BiLSTM contribute to additional performance enhancements; and (3) pretraining on analogous tasks aids in reducing the MAE. This paper establishes a state-of-art benchmark for future researches based on the new released dataset. Subsequent studies aims to further advance the capabilities of LLMs in detecting boundaries within mixed texts.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12763–12771.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Shruti Jadon. 2020. A survey of loss functions for semantic segmentation. In *2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)*, pages 1–7. IEEE.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2022. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *arXiv preprint arXiv:2212.10341*.

Daniel Martens and Walid Maalej. 2019. Release early, release often, and watch your users' emotions. *arXiv preprint arXiv:1906.06403*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector. *arXiv preprint arXiv:2310.06202*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guanliang Chen. 2023. Towards automatic boundary detection for human-ai hybrid essay in education. *arXiv preprint arXiv:2307.12267*.

Team NP_PROBLEM at SemEval-2024 Task 7: Numerical Reasoning in Headline Generation with Preference Optimization

Pawan Kumar Rajpoot *
SCB DataX, Thailand
pawankumar.rajpoot@data-x.ai

Nut Chukamphaeng *
SCB DataX, Thailand
nut.chukamphaeng@data-x.ai

Abstract

While large language models (LLMs) exhibit impressive linguistic abilities, their numerical reasoning skills within real-world contexts remain under-explored. This paper describes our participation in a headline-generation challenge by Numeval at Semeval 2024, which focused on numerical reasoning. Our system achieved an overall top numerical accuracy of 73.49% on the task. We explore the system's design choices contributing to this result and analyze common error patterns. Our findings highlight the potential and ongoing challenges of integrating numerical reasoning within large language model-based headline generation.

1 Introduction

The capacity to understand and manipulate numerical information within natural language text is essential for various NLP applications. Tasks such as news summarization, report generation, and the creation of data-driven narratives increasingly rely on the accurate interpretation and generation of numerical expressions. SemEval 2024 Task 7 (Chen et al., 2024) addresses these challenges through two intriguing subtasks: numerical headline generation and numerical headline number fill-in-the-blanks.

Generating numerical headlines necessitates models capable of synthesizing a succinct and attention-grabbing title that accurately reflects a news article's core numerical quantities and trends. Conversely, the fill-in-the-blanks subtask tests the model's ability to comprehend numerical relationships and infer the missing value to complete a provided headline. These tasks present a complex intersection of numerical reasoning and natural language generation/understanding.

Existing text generation and numerical understanding work often leverage sequence-to-sequence architectures and specialized pre-trained language models. However, SemEval 2024 Task 7's emphasis on numerical reasoning within headlines creates a distinct demand for techniques capable of accurately grounding representations of numbers and quantities within the linguistic context. This paper describes our approach to SemEval 2024 Task 7. We worked on both tasks separately and created two separate models. We used techniques such as parameter-efficient fine-tuning of large language models and then doing Direct Preference Optimization on top to align models better.

The remainder of this paper proceeds as follows. Section 3 reviews related work in numerical reasoning and headline generation. Section 4 details our models and methodology. Section 4 presents our experimental evaluation of the SemEval 2024 Task 7 dataset and includes a thorough analysis of our results. Finally, Section 5 summarizes our findings and outlines potential future research directions.

2 Background and Related Work

Headline generation within NLP has a rich history, evolving from early extractive techniques towards modern abstractive generation methods. Initial extractive approaches primarily focused on selecting the most salient sentences from the source document to compose the headline (Dorr et al., 2003) (Erkan and Radev, 2011; Mihalcea and Tarau, 2004). These methods offered interpretability but lacked the fluency and novelty often desired in generated headlines. The advent of deep learning and sequence-to-sequence models enabled abstractive headline generation, empowering models to synthesize new phrases and expressions (Rush et al., 2015; Nallapati et al., 2016). Attention mechanisms (Bahdanau et al., 2015) proved pivotal in aligning source text and headline generation. Recent advancements in Generative AI have led to significant improvements in this field with state-of-the-

*Equal Contribution

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 716–720

art results (Zhang et al., 2019). *GSum* (Dou et al., 2020), for example, initially performs extractive summarization and then incorporates the extractive summaries into the input for abstractive summarization. *SEASON* (Wang et al., 2022) adopts a dual approach, learning to predict the informativeness of each sentence and using this predicted information to guide abstractive summarization Notably, most of these works focus on the selection of words and the structure of sentences.

3 Numeval

Numeval is part of Semeval 2024; the task we focused on and worked on requires models to generate concise and informative headlines that accurately reflect the core numerical information in news articles. Systems must demonstrate an understanding of how numbers convey meaning and should prioritize the most relevant numerical aspects for inclusion in the headline.

Subtask 1: Numerical Reasoning - models are required to compute the correct number to fill the blank in a news headline.

Subtask 2: Abstractive Headline Generation - models must construct a headline based on the provided news; this headline should incorporate the numerical reasoning within. The organizers released

News:
At least **30** gunmen burst into a drug rehabilitation center in a Mexican border state capital and opened fire, killing **19** men and wounding **four** people, police said. Gunmen also killed **16** people in another drug-plagued northern city. The killings in Chihuahua city and in Ciudad Madero marked one of the bloodiest weeks ever in Mexico and came just weeks after authorities discovered **55** bodies in an abandoned silver mine, presumably victims of the country's drug violence. More than **60** people have died in mass shootings at rehab clinics in a little less than **two** years. Police have said **two** of Mexico's **six** major drug cartels are exploiting the centers to recruit hit men and drug smugglers, ...

Headline (Question): Mexico Gunmen Kill ____

Answer: 35

Annotation: Add(19,16)

Figure 1: Numeval Task 3-1 examples.

a novel dataset designed to facilitate research on numeral-aware headline generation. The NumHG (Huang et al., 2023) dataset addresses the issue of inaccurate numeral generation in headline creation. It provides over 27,000 news articles with detailed annotations designed to facilitate the development of models that accurately understand and summarize numerical information. For subtask 1, each data point has an answer operator added, which signifies how the numerical answer is obtained, which includes Copy (direct retrieval), Trans, Span, Round, Paraphrase, Add, Subtract, Multiply, and Divide. Meanwhile, subtask 2 requires the model

News:
(Apr 18, 2016 1:02 PM CDT) Ingrid Lyne, the Seattle mom allegedly murdered while on a date, left behind three daughters—and a GoFundMe campaign set up to help the girls has raised more than $222,000 so far, Us reports. A friend of the family set up the campaign, and says that all the money raised will go into a trust for the girls, who are ages 12, 10, and 7. Lyne's date was charged with her murder last week.

Headline: $222K Raised for Kids of Mom Dismembered on Date

Figure 2: Numeval SubTask 2 examples.

to generate complete headlines from given news content.

4 Methodology

4.1 subtask 1

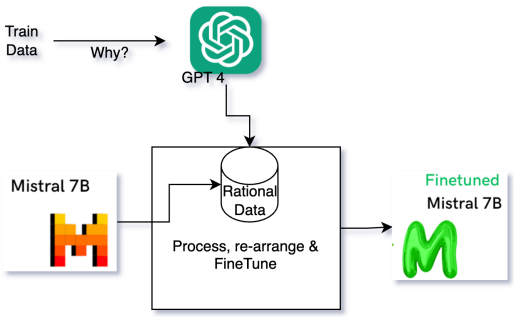


Figure 3: Overall Stage 1 train subtask 1

For this subtask, we start by passing the passage and question in the blank statement with an aligned answer to GPT 4 and ask it to generate a rationale for this given answer. The below prompt is used in this process.

```
PASSAGE: ARTICLE-HERE
QUESTION: FILL_IN_THE_BLANKS-HERE
WHY ANSWER TO THIS IS ANSWER-HERE ?
EXPLAIN\nRESULT:
```

Once we get the reasoning from this module, we restructure the training data in the following manner.

```
PASSAGE: ARTICLE-HERE
QUESTION: FILL_IN_THE_BLANKS-HERE
```

WHY ANSWER TO THIS IS ANSWER-HERE ?
REASON: GPT_RATIONALE_HERE

We use this to train our main model for subtask 1.

4.2 subtask 2

1) **Numerical Reasoning**: The model must demonstrate fluency in numerical calculations.
2) **Headline Matching**: Generated headlines must stylistically align with the data.

We worked on an end-to-end solution leveraging a Large Language Model (LLM) to address these challenges. First, we fine-tune the LLM to enhance its mathematical reasoning capabilities. This approach targets accurately interpreting numerical data and producing suitable headlines.

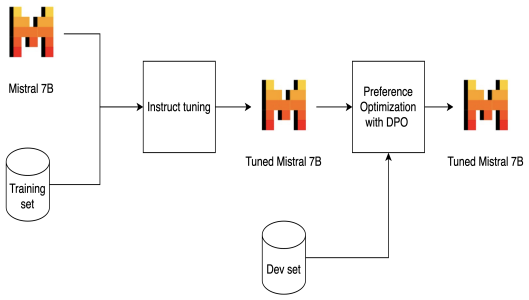


Figure 4: Overall workflow for subtask 2

4.3 Fine-tuning

For both tasks separately, we selected Mistral-7B (Jiang et al., 2023) as our trained LLM based on its strong performance on diverse benchmarks, including those focused on numerical reasoning benchmarks like GSM8K, which suggests a solid foundation for further fine-tuning on our specific numerical headline generation task.

As illustrated in Figure 5, Mistral-7B achieve a lower fine-tuning loss than BART (Lewis et al., 2019), a state-of-the-art text summarization model. This reinforces its suitability for our task.

For efficient fine-tuning, we employ Parameter Efficient Fine Tuning (PEFT). Due to memory constraints, we employed 4-bit QLoRA(Dettmers et al., 2023) quantization (cite reference) with a rank of 128 and an alpha of 256. This quantization technique was applied specifically to the self-attention Query, Key, and Value matrices along with the Linear layers of the model. To optimize the process, we used gradient accumulation (steps=2), a paged 32-bit Adamw optimizer, a cosine learning rate schedule (LR=2e-5), a decay rate of 0.01, and a short 5-step warmup period. The entire fine-tuning process was facilitated using the axolotl library. This technique reduces the model's memory footprint while minimizing performance degradation. This is particularly advantageous when working with LLMs.

Prompt template

> Given the news article, please write an appropriate headline
> {news content}
>
> Headline:

4.4 Direct Preference Optimization (DPO)

For both subtasks, we further aligned our fine-tuned models to learn better using the dev set. We did not use the dev data split in the first train stage. While aligning, we used dev data to first run through the model. We realigned the fine-tuned model with incorrect outcome results, that is, the dev results where the predicted number in the generated headline was incorrect or rejected. We still use the rationale (for subtask 1) for DPO, while DPO training for the second subtask only contains the predicted headline(wrong/rejected) and the correct/choosen headline. One example of the DPO train data for subtask 1 is below.

> PASSAGE:
> Stocks made gains today, extending a winning streak into its fourth day, MarketWatch reports. Merck rose 12.5% on announcement of its merger with Schering-Plough, while General Motors built on recent gains with a 22.9% jump. The Dow closed up 53.92 at 7,223.98.,
> QUESTION:
> "Dow Up ____, Gains 9% for Week
> **Choosen**:54 REASON: rounding off to nearest integer
> **Rejected**: 53.92 REASON:copy from text

To align the generated fill-in-the-blanks/headlines style with the target dataset, we utilize the Direct Preference Optimization (DPO) alignment technique (Rafailov et al., 2023). Direct Preference Optimization (DPO) is a novel approach for aligning large language models (LLMs) with human preferences. Unlike traditional methods that rely on reward models and reinforcement learning, DPO leverages human feedback through preferred and dispreferred outputs to directly train the LLM. This simplifies the training process and avoids the complexities of reward model design. This helps us

Table 1: Automatic evaluation results subtask 2.

	Num Acc.			ROUGE			BERTScore			MoverScore
	Overall	Copy	Reasoning	1	2	L	P	R	F1	
ClusterCore	38.233	51.571	13.942	33.467	11.837	28.927	31.876	42.232	37.026	56.405
Noot Noot	38.393	57.481	3.6331	31.47	11.139	27.284	25.389	43.977	34.539	55.559
Infrrd.ai	65.840	68.354	61.263	46.789	22.36	42.095	51.005	47.260	49.134	59.731
hinoki	62.347	66.284	55.177	43.072	19.719	38.999	47.223	43.444	45.342	58.711
Challenges	72.956	**82.170**	56.176	31.220	12.235	26.859	19.530	47.559	33.132	55.362
NCL_NLP	62.122	65.536	55.904	43.506	19.388	38.878	46.402	45.039	45.734	58.861
YNU-HPCC	69.044	73.018	61.807	**48.852**	**24.681**	**44.175**	**51.553**	**50.095**	**50.381**	**60.551**
NoNameTeam	55.715	57.681	52.134	40.646	17.261	35.745	44.256	40.387	42.324	57.736
np_problem (ours)	**73.487**	76.908	**67.257**	39.816	17.577	34.339	27.800	48.557	37.816	57.024

fine-tune the model beyond numerical correctness to produce stylistically suitable headlines.

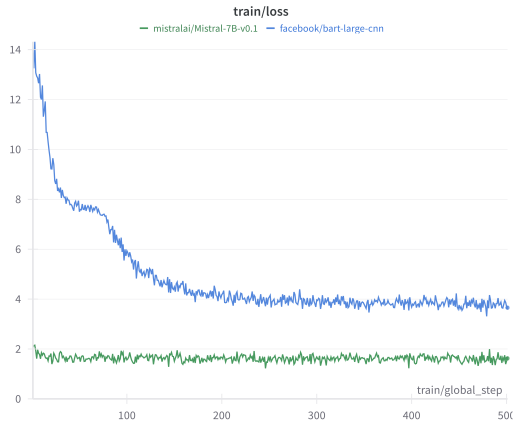


Figure 5: Comparing fine-tuning losses between BART and Mistral-7B

5 Results

For subtask 1, we evaluate our model on two fill-in-the-blank types. One accuracy on Copy, where the answer is directly copied from the article. Second is Reasoning, which includes different reasoning techniques such as addition, subtraction, multiplication, and paraphrasing.

Table 2: Accuracy on tasks type subtask 1

Copy	Reasoning
0.922	0.784

For comparative semeval two-stage evaluations, we scored 0.89 in the open and 0.86 in the hidden stage, respectively.

Table 3: Comparitive num accuracy on subtask 1

Team	Open-Score	Hidden-Score
CTYUN-AI	**0.95**	**0.95**
zhen qian	0.94	0.94
YNU-HPCC	0.93	0.94
NP-Problem(ours)	0.89	0.86

For subtask 2 we performed best in numerical accuracy overall and reasoning scores, we out-shined in reasoning accuracy as difference between 1st and 2nd rank was around 7 points. We open-source our final models on Huggingface [1].

6 Limitations

Since our method is based on 7B LLM, our performance is capped by this model's ability to draw rationale for the numerical reasoning. This limitation is in line with the hardware resource as well; We used an RTX 4090 24GB GPU-based machine for our work, which can load and fine-tune the models with upto 7-10 B parameters as well.

Conclusion

We present a modular solution to the numeral problem with an alignment module to increase the model's ability to understand numerical reasoning across both tasks. We thank the organizing committee of SemEval-2024, along with the task-setting team of Numeval, for allowing us to work on this problem.

[1] https://huggingface.co/lingjoor/numeval-task7-1, https://huggingface.co/lingjoor/numeval-task7-2

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 1–8.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2020. Gsum: A general framework for guided neural abstractive summarization. *CoRR*, abs/2010.08014.

Günes Erkan and Dragomir R. Radev. 2011. Lexrank: Graph-based lexical centrality as salience in text summarization. *CoRR*, abs/1109.2128.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685.

Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. 2022. Salience allocation as guidance for abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6094–6106, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

OPDAI at SemEval-2024 Task 6: Small LLMs can Accelerate Hallucination Detection with Weakly Supervised Data

Chengcheng Wei, Ze Chen, Songtan Fang, Jiarong He and **Max Gao**
Interactive Entertainment Group of Netease Inc., Guangzhou, China
{weichengcheng, jackchen, fangsongtan, gzhejiarong, jgao}@corp.netease.com

Abstract

This paper mainly describes a unified system for hallucination detection of LLMs, which wins the second prize in the model-agnostic track of the SemEval-2024 Task 6, and also achieves considerable results in the model-aware track. This task aims to detect hallucination with LLMs for three different text-generation tasks without labeled training data. We utilize prompt engineering and few-shot learning to verify the performance of different LLMs on the validation data. Then we select the LLMs with better performance to generate high-quality weakly supervised training data, which not only satisfies the consistency of different LLMs, but also satisfies the consistency of the optimal LLM with different sampling parameters. Furthermore, we finetune different LLMs by using the constructed training data, and finding that a relatively small LLM can achieve a competitive level of performance in hallucination detection, when compared to the large LLMs and the prompt-based approaches using GPT-4.

1 Introduction

The emergence of Large Language Models (LLMs)(Zhao et al., 2023) has sparked a significant transformation in the field of Natural Language Processing (NLP), ushering in a new era of unparalleled advancements in text generation and comprehension. This revolutionary technology has elevated the capabilities of AI systems, enabling them to perform complex reasoning and problem-solving tasks with remarkable proficiency(Zhao et al., 2023). At the heart of this transformation lies the LLMs' ability to compress vast amounts of knowledge into neural networks, effectively turning them into versatile agents capable of engaging in natural language conversations with humans(Hadi et al., 2023). This has broadened the scope of AI applications beyond traditional domains such as chatbots and virtual assistants, into areas previously thought to be the exclusive domain of humans, particularly those involving creativity and expertise. LLMs are not only limited to language-related tasks but can also function as generalist agents, collaborating with external systems, tools, and models to achieve a wide range of objectives set by humans(Triguero et al., 2024).

However, recent advancements in research have uncovered a concerning weakness: their proneness to hallucinate content across a range of applications(Ji et al., 2023). Hallucination is defined as the generation of information that either conflicts with established sources or cannot be substantiated by available knowledge. The occurrence of hallucination in LLMs poses a significant threat to their practical deployment. While prior works have delved into the roots of hallucination within specific, smaller-scale language models and tasks, there is still a notable gap in understanding the exact nature and prevalence of content that LLMs are likely to hallucinate(Cui et al., 2024; Chang et al., 2023).

To address this challenge, we implements a unified system for hallucination detection of LLMs, when there is no labeled training data. This system comprises five parts: *Base Model Selection*, *Prompt Engineering*, *Weakly-supervised Data Generation*, *SFT* and *Ensemble Learning*. We first verify the performance of different base LLMs on this task. Then we select the best LLMs and prompt is optimized to improve the performance. And weakly-supervised dataset is generated by using the selected LLMs. For further improvement, SFT is done based on the constructed dataset and ensemble learning is adopted.

2 Related Work

In this section, we will introduce other work related to the subsequent methods.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 721–729

Mixture of Experts(MoE)(Jacobs et al., 1991) is an AI technique that involves a group of specialized models (experts) being coordinated by a gating mechanism to address various aspects of the input space, in order to optimize performance and efficiency. This approach capitalizes on the idea that an ensemble of weaker language models, each focusing on specific tasks, can yield more precise results, similar to traditional ML ensemble methods. However, it introduces a novel concept of dynamically routing the input during the generation process. In the subsequent methods, we will conduct comparative experiments using the intelligent model base of MoE. This paper(Chen et al., 2022) integrates POS information and word semantic representation using an MoE approach.

Model ensembling combines the predictions from multiple models together. Traditionally this is done by running each model on some inputs separately and then combining the predictions. However, if the candidate models have the same architecture, they can be combined together to create a new model for prediction. Many surveys systematically elucidate the basic concepts of ensemble learning and various methods, including model fusion and model voting(Krawczyk et al., 2017; Dong et al., 2020; Yang et al., 2023).

LoRA (Low-Rank Adaptation of Large Language Models) (Hu et al., 2021) is a widely used and lightweight training technique that markedly reduces the number of trainable parameters. It functions by adding a smaller set of new weights to the model and training only these. As a result, training with LoRA is notably faster, more memory-efficient, and yields smaller model weights (a few hundred MBs), which are more manageable for storage and sharing. Some other methods(Ye et al., 2023; Wang et al., 2023; Chen et al., 2023) have been developed to improve LoRA.

Chain-of-Thought Prompting(CoT)(Wei et al., 2022) is an emerging application of language model technology. The core idea of this method is to encourage the model to not only generate the final answer but also gradually demonstrate its reasoning and the process of reaching conclusions. Subsequent work(Zhou et al., 2022) has applied the idea of CoT, breaking down problems into a series of sub-problems, allowing the model to reason step by step and ultimately provide the correct answer. And another work(Wang et al., 2022) introduces a method called "self-consistency" to further enhance the effectiveness of CoT Prompting. By generating multiple reasoning paths and selecting the most consistent answer, the model can reduce errors and improve the accuracy of reasoning.

3 Task Description

trial data	unlabled train data	validation data	test data
80	60000	1000	3000

Table 1: Dataset provided by SHROOM

SHROOM(Mickus et al., 2024) asked participants to perform binary classification to identify cases of fluent overgeneration hallucinations in two different setups: model-aware and model-agnostic tracks. And three different NLG tasks: definition modeling (DM), machine translation (MT) and paraphrase generation (PG) are covered in both tracks. In model-aware track, the model information is provided. The provided development and test sets include binary annotations from a minimum of five different annotators, along with a majority vote gold label. Table 1 gives an overview of the provided dataset.

4 Methodology

Figure 1 shows the overview of our approach. Our method consists of five main steps. First of all, multiple LLMs are compared on the hallucination detection validation dataset and among which we select the best base model. The LLMs selected in the first step will be utilized in the subsequent steps 2, 3, and 4. In the second step, we designed a Prompt Engineering module consisting of three sub-modules: few-shot prompting, instruction optimization, and the utilization of Chain-of-Thought. The subsequent experiments will demonstrate that prompt engineering significantly enhances the capabilities of the base model.

Then moving on to the Label Generation step, we apply the Prompt Engineering module to the selected best LLMs. We make predictions on the unlabeled training set and ensure the inference consistency among multiple LLMs as well as the inference consistency under different inference parameters of individual LLM. We also ensure label balance in this process.

The following steps are Model Training and Ensemble Learning. We utilize SFT based on the constructed dataset using the LoRA (Hu et al., 2021) method. Finally, we select a few top-performing

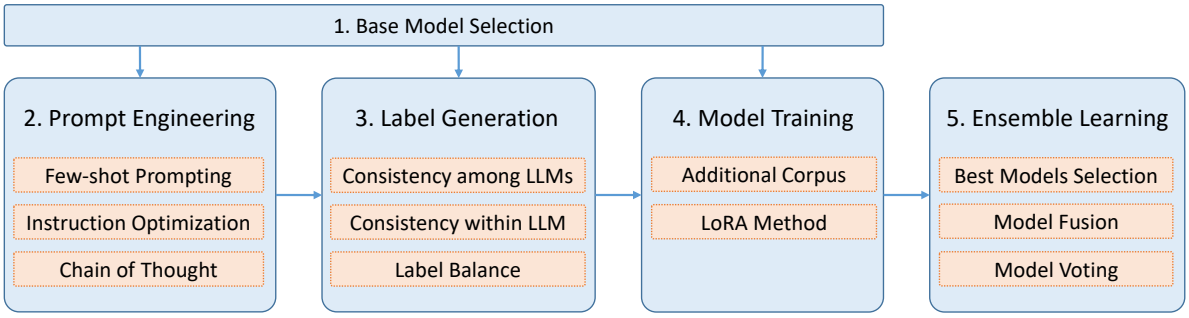


Figure 1: Overview of our proposed method. The method consists of five main steps, each of which comprises several modules that we have designed for SemEval-2024 Task 6.

models and perform fusion at both the model weight level(model fusion) and the model prediction probability level(model voting) in order to seek better performance.

4.1 Prompt Engineering

After experimenting with multiple large-scale language models, we select the best-performing 14B *Mixtral_7Bx2_MoE* as the base model for prompt engineering to achieve better results.

Few-shot prompting. Few-shot prompting can be used as a technique to enable in-context learning where we provide demonstrations from the provided *trial data* in the prompt to steer the base model to better performance. Considering that the trial data contains with labels, we randomly sample the specific task's datapoints for different tasks, ensuring an equal number of data points for "hallucination" and "not hallucination" examples, to serve as few-shot examples for each respective task.

Optimizing the instruction. There are some deficiencies with the instruction introduced in Section 5.1, and we name this version of instruction as the *naive* version. The most obvious issue is that the naive instruction does not include the descriptions of the DM, MT, and PG tasks. The desired task description includes the task definition and all known useful information, rather than just focusing on the sentence and context. We design different instructions for different tasks, which can be found in the Appendix A. In this way, we can append additional information to the prompt to assist the LLM in better understanding the problem.

Chain-of-thought prompting. Chain-of-thought (CoT) prompting(Wei et al., 2022) is a recently developed method that encourages the language model to explain its reasoning. We combine the aforementioned few-shot prompting, developed instruction, and CoT to utilize them together to further enhance the capability of the base model.

4.2 Label Generation and Weakly-SFT

After improving the performance of the base model using prompt engineering, we use the optimal settings to infer on unlabeled training data and obtain weakly supervised labels. These weakly supervised labels are then used to finetune the base model.

Inference consistency in generating labels. During the process of inferring weakly supervised labels for the unlabeled training data, we placed a great emphasis on both the consistency of inference across different LLMs and the consistency of inference within the same LLM but with different parameter settings. To achieve this, we carefully selected several sets of top-performing base models. Leveraging the prompt engineering techniques mentioned earlier, we conducted inference on the same model using various parameter configurations. Subsequently, we handpicked the data points with consistent inferences across different parameter settings to establish the final inference results for that particular LLM. Additionally, we applied a filtering process to the inference results obtained from different LLMs, ensuring that only datapoints with consistent inferences were retained. Through these rigorous steps, we ensured that our generated weakly supervised labels for the training set exhibits robustness to both the choice of LLM base and the specific sampling parameters employed. Finally, we used sampling techniques to balance the data volume of the two categories.

Fine-tuning LLMs. The weak supervision generated by the base model is applied to guide models of equal or smaller scale. The LLMs undergo fine-tuning using the LoRA approach, a popular and lightweight training technique that effectively

Model Name	Model Size	Model-agnostic track		Model-aware track	
		acc	*rho*	*acc*	*rho*
Mistral-7B-Instruct-v0.2-GGUF	7B	0.649	0.380	0.707	0.461
Mistral-7B-Instruct-v0.2	7B	0.655	0.375	0.705	0.468
Mixtral_7Bx2_MoE	14B	**0.747**	0.518	0.764	0.475
Mixtral-8x7B-Instruct-v0.1	46.7B	0.723	0.526	0.745	0.552
Nous-Hermes-2-Mixtral-8x7B-DPO	46.7B	0.741	0.607	**0.766**	0.614
Nous-Hermes-2-SOLAR-10.7B	10.7B	0.725	0.592	0.722	0.588
SOLAR-10.7B-Instruct-v1.0	10.7B	0.737	0.438	0.747	0.381
SauerkrautLM-SOLAR-Instruct	10.7B	0.733	0.418	0.752	0.368
Sakura-SOLAR-Instruct-DPO-v2	10.7B	0.733	0.426	0.745	0.357

Table 2: The performance of different-sized LLMs on the validation set. The competition includes two tracks: model-agnostic track and model-aware track. For each track, both prediction accuracy(*acc*) and Spearman's Rho value(*rho*) are provided.

reduces the number of trainable parameters. Despite this reduction, the fine-tuned models maintain comparable training results to those of the full parameter models. The best checkpoint model files are selected from the validation set during this fine-tuning process.

4.3 Ensemble Learning

We also propose an ensemble learning approach for performance improvement, utilizing fusion strategies at both the model level and the inference level.

Model fusion. MergeKit[1] is a toolkit designed for merging trained language models. We carefully selected a few high-accuracy models and utilized MergeKit to perform model fusion using the SLERP (Shoemake, 1985), TIES (Yadav et al., 2023) and linear (Wortsman et al., 2022) methods. Traditionally, model merging often resorts to weight averaging which, although straightforward, might not always capture the intricate features of the models being merged. The SLERP technique addresses this limitation, producing a blended model with characteristics smoothly interpolated from both parent models, ensuring the resultant model captures the essence of both its parents. Meanwhile, the TIES method is proposed to resolve interference issues by resetting parameters, resolving sign conflicts, and merging only compatible parameters. TIES outperforms many existing methods across diverse settings, emphasizing the importance of addressing interference in model merging for enhanced performance and versatility.

[1] https://github.com/arcee-ai/mergekit

Model Voting. In addition to the model-level fusion, we also explored fusion at the probability level of model generation, which can be understood as a form of model voting. We selected another group of highly accurate candidate models and performed linear fusion at the probability level. Specifically, we calculate the weighted summation of the probability values on "existing hallucination" predicted by different candidate models for different tasks. By tuning the linear weight combination, we are able to determine the optimal combination of weights for each task. Finally, combining different tasks together yields the final fusion result. In this way, we implement weighted voting of models at the inference result level.

5 Results and Analysis

In this section, we will present a series of experiments to illustrate the effectiveness of our method.

5.1 Baseline

To begin with, our initial step entails presenting the basic performance of LLMs of varying sizes on the validation set. Subsequently, we will delve into an analysis of the LLMs' capabilities in detecting hallucinations in the given task.

Throughout the experiments, we ensure the generation hyperparameters remain consistent across all LLMs. Additionally, the instruction utilized for detecting hallucinations is sourced from the official *participant_kit*. This version of the instruction is referred to as the "naive instruction" and can be located in Appendix A for reference.

Table 2 illustrates our evaluation of LLMs from both *Mistral* and *SOLAR* families, considering vary-

few-shot	inst.	agnostic_acc	aware_acc
2-shot	naive	0.745	0.774
	ours	**0.770**	**0.806**
4-shot	naive	0.762	0.772
	ours	**0.782**	**0.806**
6-shot	naive	0.764	0.774
	ours	**0.772**	**0.804**
8-shot	naive	0.762	0.772
	ours	**0.774**	**0.804**

Table 3: Our proposed instruction exhibits overall superior accuracy compared to the naive version on the validation set. We applied few-shot prompting in all of the aforementioned experiments. The term "inst." stands for "instruction".

few-shot	CoT	agnostic_acc	aware_acc
2-shot	w/o	0.770	**0.806**
	with	0.770	0.792
4-shot	w/o	**0.782**	**0.806**
	with	0.766	0.796
6-shot	w/o	0.772	0.804
	with	**0.774**	**0.806**
8-shot	w/o	0.774	0.804
	with	**0.792**	0.804

Table 4: CoT demonstrates an improved capability in hallucination detection when provided with a larger number of demonstrations in few-shot prompting and utilizing our proposed instruction. The results are on the validation set.

ing sizes and variants, on the validation set. In general, larger models tend to yield better results. For instance, within the *Mistral*-family, the accuracy and Spearman's Rho value of the 7B model are comparatively lower than those of larger models, a trend observed in both the model-agnostic and model-aware tracks. Furthermore, LLMs of the same size exhibit diverse results in hallucination detection tasks owing to distinct fine-tuning methods employed. This observation holds true in our experiments with the *SOLAR*-family, emphasizing the impact of fine-tuning on performance.

There is a noteworthy observation in Table 2. The medium-sized 14B *Mixtral_7Bx2_MoE* model achieves comparable accuracy to the larger-sized 46.7B models in both the model-agnostic and model-aware tracks. This suggests that the fine-tuning approach and training corpus of the 14B model are well-suited for the hallucination detection task. Furthermore, the 14B model outperforms the 46.7B model in terms of inference speed and training cost. As a result, in the subsequent section, we will further enhance the effectiveness of the 14B model through prompt engineering.

5.2 Performance Improvement

In this section, our focus is on enhancing the accuracy of the 14B *Mixtral_7Bx2_MoE* model through prompt engineering methods.

Few-shot prompting. A few-shot prompting approach is applied by randomly selecting an equal number of positive and negative samples as demonstrations for task definition modeling (DM), machine translation (MT), and paraphrase generation (PG). In the experimental setup, we use 2, 4, 6, and 8 examples for few-shot prompting on the *Mixtral_7Bx2_MoE* model, while keeping the generation hyperparameters consistent with the experiments in Table 2. The accuracy of the few-shot prompting strategy is shown with the *inst.=naive* setting in Table 3, where we observe that experiments with 4, 6, and 8 shots perform better than the zero-shot baseline(acc is 0.747 in Table 2) in the model-agnostic track, and all the few-shot settings experiments achieve better results than the zero-shot baseline(acc is 0.764 in Table 2) in the model-aware track.

Optimizing the instruction. As discussed in Section 4.1, the naive instruction provided by the competition organizers has some limitations. To overcome these limitations, we enhanced the instructions by incorporating task-specific background knowledge and multidimensional information, taking into account the unique characteristics of each task. The improved instructions, as demonstrated with the *inst.=ours* setting in Table 3, yield better performance compared to using the initial naive instruction. Notably, in the 2-shot setting, both tracks exhibited an improvement of over 2 percentage points by leveraging our proposed instructions.

Chain of thought prompting. We adopt the CoT approach, after generating reasons for the presence or absence of hallucinations in the trial data. The experimental results of CoT are presented in Table 4, which indicates that CoT exhibits higher efficacy in the few-shot scenario when there are more demonstrations accessible.

Model Name	Model Size	Model-agnostic track		Model-aware track	
		acc	*rho*	*acc*	*rho*
Mistral-7B-Instruct-v0.2	7B	**0.806**	0.708	0.790	0.699
Nous-Hermes-2-SOLAR-10.7B	10.7B	0.764	0.690	0.806	0.714
SOLAR-10.7B-Instruct-v1.0	10.7B	0.772	0.703	0.810	0.717
Mistral-7B-Instruct-v0.2-2x7B-MoE	12.8B	0.790	0.725	**0.814**	0.698
Mixtral_7Bx2_MoE	14B	0.780	0.675	0.796	0.657
raw: Mixtral_7Bx2_MoE	14B	0.792	0.707	0.804	0.690

Table 5: The performance of weakly-supervised fine-tuning on the validation set. Smaller LLMs perform better than the 14B supervisor. The last line indicates the performance of Mixtral_7Bx2_MoE without SFT.

Method	agnostic_acc
Mistral-7B-Instruct-v0.2-sft-v1	0.804
Mistral-7B-Instruct-v0.2-sft-v2	0.798
Mistral-7B-Instruct-v0.2-sft-v3	0.808
Linear-merged model	**0.814**
SLERP-merged model	**0.814**
TIES-merged model	**0.814**

Table 6: Model fusion results for the model-agnostic track on the validation set. The merged models outperform any individual model in terms of accuracy.

Method	agnostic_acc
Mistral-7B-Instruct-v0.2-sft-v4	0.810
Mistral-7B-Instruct-v0.2-sft-v5	0.812
Mistral-7B-Instruct-v0.2-sft-v6	0.812
Mistral-7B-Instruct-v0.2-sft-v7	0.814
voting result	**0.834**

Table 7: Model Voting results for the model-agnostic track on the validation set. The voted results outperform any individual model in terms of accuracy.

5.3 Weakly-supervised Fine-tuning

As mentioned earlier, we enhance the accuracy of hallucination detection by selecting the best baseline model and incorporating additional prompt engineering techniques. Building upon this, we leverage weak supervision by labeling the unlabeled training data for training. Subsequently, the LLMs are fine-tuned using the generated labels to further augment the capability of hallucination detection.

Generating weak supervision for training data. Utilizing the 14B *Mixtral_7Bx2_MoE* model as a foundation, we incorporate few-shot prompting, our proposed instruction, and the CoT strategy to create a supervision model known as the '8-shot' setting, as mentioned in Table 4. This approach is applied to hallucination detection across 60,000 datapoints from both the model-agnostic and model-aware tracks, ensuring a balanced distribution of categories. Additionally, we introduce multiple optimal models, as discussed in Section 4, to ensure consistent inference across multiple models and maintain inference consistency within the same model but with different inference parameters.

Fine-tuning LLMs. The experimental results of weakly-supervised fine-tuning are presented in Table 5, which demonstrates that smaller models can effectively learn from the weak supervision provided by the 14B model. In some cases, these smaller models even outperform the 14B model in terms of accuracy. Notably, the 14B model fails to surpass the performance of equivalently-sized supervisor even when multiple hyper-parameter settings are employed. A comparison between lora training and full-parameter training reveals that the lora-style training yields superior results. Further details can be found in Appendix C.

5.4 Ensemble Learning

In addition to fine-tuning LLMs with weak supervision labels as mentioned in the previous section, we first combine different model checkpoints by the MergeKit tool and then perform model voting strategy to enhance performance.

Model fusion. We implement model merging using different modes of MergeKit, *i.e.*, SLERP, TIES and Linear, in our study. Taking the accuracy optimization of the model-agnostic track as an example, let's begin by selecting three highly capable candidate models. These models are fine-tuned versions of the 7B *Mistral-7B-Instruct-v0.2*, and they are different checkpoints from the same training task. The detailed training setting can be

found in Appendix B.2. By utilizing model fusion techniques, we can achieve a maximum accuracy of 0.814 with the newly merged models. For a detailed overview of the experiments on the model-agnostic track, refer to Table 6.

Model Voting. We also validate the effectiveness of weighted voting at the inference result level. We select the top-performing models from the weakly supervised fine-tuning and model fusion phases. By calculating weighted predictions based on their predicted probabilities, we infer the presence of hallucinations. Table 7 presents the details of the voting experiments on the model-agnostic track. Using the same method, we achieve an accuracy of 0.818 on the model-aware track as well. The SFT models are merged models from different training experimental setups, and the detailed training parameters can be found in Appendix B.3.

Method	agnostic_acc	aware_acc
baseline	0.697	0.745
GPT-4	0.741	0.756
our method	**0.836**	**0.805**

Table 8: Comparison of methods on the test set.

We compared the baseline provided by competition organizers, GPT-4, and our proposed method on the test set in Table 8. It is evident that our proposed method outperforms other mehotds, showcasing a significant enhancement in performance.

6 Conclusion

In this paper, we present a unified system for hallucination detection with LLMs when there is no labeled dataset, which wins the 2nd place with an accuracy score of 0.836 in the model-agnostic track and the 4th place with an accuracy score of 0.8053 in the model-aware track. To begin with, we generate high-quality weakly-supervised dataset by using large-sized LLMs with prompt engineering and few-shot learning. Then we perform weakly-supervised fine-tuning based on the constructed dataset with different LLMs. Our experiments yield several noteworthy findings:

(1) The quality of the weakly-supervised dataset we construct has a direct impact on the performance of the models in this task. To ensure high-quality training data, we employ multiple large LLMs in the construction process.

(2) Relatively small LLMs can deliver competitive performance in this task when trained on the constructed dataset. However, the performance of small LLMs drops dramatically without fine-tuning.

(3) Using the *MergeKit* tool for model fusion proves to be an effective technique in boosting the performance of hallucination detection.

(4) Employing the model voting method leads to improved performance compared to using a single model alone.

References

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Ze Chen, Kangxu Wang, Zijian Cai, Jiewen Zheng, Jiarong He, Max Gao, and Jason Zhang. 2022. Using deep mixture-of-experts to detect word meaning shift for TempoWiC. In *Proceedings of the The First Workshop on Ever Evolving NLP (EvoNLP)*, pages 7–11, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Tianyu Cui, Yanling Wang, Chuanpu Fu, Yong Xiao, Sijia Li, Xinhao Deng, Yunpeng Liu, Qinglin Zhang, Ziyi Qiu, Peiyang Li, et al. 2024. Risk taxonomy, mitigation, and assessment benchmarks of large language model systems. *arXiv preprint arXiv:2401.05778*.

Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Bartosz Krawczyk, Leandro L Minku, João Gama, Jerzy Stefanowski, and Michał Woźniak. 2017. Ensemble learning for data stream analysis: A survey. *Information Fusion*, 37:132–156.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Ken Shoemake. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254.

Isaac Triguero, Daniel Molina, Javier Poyatos, Javier Del Ser, and Francisco Herrera. 2024. General purpose artificial intelligence systems (gpais): Properties, definition, taxonomy, societal implications and responsible governance. *Information Fusion*, 103:102135.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yiming Wang, Yu Lin, Xiaodong Zeng, and Guannan Zhang. 2023. Multilora: Democratizing lora for better multi-task learning. *arXiv preprint arXiv:2311.11501*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yongquan Yang, Haijun Lv, and Ning Chen. 2023. A survey on ensemble learning under the era of deep learning. *Artificial Intelligence Review*, 56(6):5545–5589.

Zhengmao Ye, Dengchun Li, Jingqi Tian, Tingfeng Lan, Jie Zuo, Lei Duan, Hui Lu, Yexi Jiang, Jian Sha, Ke Zhang, et al. 2023. Aspen: High-throughput lora fine-tuning of large language models with a single gpu. *arXiv preprint arXiv:2312.02515*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Instructions in Prompt Engineering

A.1 Naive version

Context: {Context}
Sentence: {Sentence}
Is the Sentence supported by the Context above?
Answer using ONLY yes or no:

A.2 Our proposed version

For the PG task, the instruction is as follows:

Given the following information related to Paraphrase Generation task:
Src: Source input sentence
Tgt: Paraphrase Generation standard answer
Hyp: Paraphrase Generation predicted answer
Please determine whether hyp contains unexpected hallucinations based on src and tgt.

Src: {Src}
Tgt: {Tgt}
Hyp: {Hyp}
Is the Hyp supported by the Src and Tgt above?
Answer using ONLY yes or no:

For the MT task, the instruction is as follows:

Given the following information related to Machine Translation task:
Src: Source input sentence
Tgt: Machine Translation standard answer
Hyp: Machine Translation predicted answer
Please determine whether hyp contains unexpected hallucinations based on src and tgt.

Src: {Src}
Tgt: {Tgt}
Hyp: {Hyp}
Is the Hyp supported by the Src and Tgt above?
Answer using ONLY yes or no:

As for the DM task, the instruction is the same as the naive version.

B Training Experiment Setup

B.1 Constructed dataset

We constructed a total of **35,600** weakly supervised samples, ensuring consistency in inference across different LLMs as well as within the same LLM but with different parameter settings.

B.2 SFT models in Table 6

The Mistral-7B-Instruct-v0.2-sft-v1, v2, and v3 models are different checkpoint models obtained from the same training setup. These models were trained on a total of 35,600 weakly-supervised data points. The training process utilized a LoRA rank of 32, a learning rate of $3e^{-5}$, and a total of 5 epochs. The training task was executed using 4 A30 GPUs. Specifically, Mistral-7B-Instruct-v0.2-sft-v1, v2, and v3 models were saved at training steps 1000, 3000, and 4000, respectively.

B.3 SFT models in Table 7

The Mistral-7B-Instruct-v0.2-sft-v4, v5, v6, and v7 models are merged models obtained from different training setups. Each model is created by merging two checkpoints from the same setup. The v4 model was trained with a LoRA rank of 32, a learning rate of $1e^{-4}$, and a total of 5 epochs. The v5 model also had a LoRA rank of 32, a learning rate of $3e^{-5}$, and a total of 5 epochs. The v6 model had a higher LoRA rank of 48, a learning rate of $3e^{-5}$, and lasted for 5 epochs. Lastly, the v7 model had a LoRA rank of 48, a learning rate of $5e^{-5}$, and a total of 5 epochs. All of these models were trained on the constructed dataset.

C Lora training VS. Full training

Method	agnostic_acc	aware_acc
lora	0.806	0.790
full	0.58	0.52

Table 9: Comparison of different training methods based on Mistral-7B-Instruct-v0.2.

SSN_ARMM at SemEval-2024 Task 10: Emotion Detection in Multilingual Code-Mixed Conversations using LinearSVC and TF-IDF

Rohith Arumugam S
Department of CSE
SSN College of Engineering
rohitharumugam2210376@ssn.edu.in

Angel Deborah S
Assistant Professor
Department of CSE
SSN College of Engineering
angeldeborahS@ssn.edu.in

Rajalakshmi S
Assistant Professor
Department of CSE
SSN College of Engineering
rajalakshmis@ssn.edu.in

Milton R S
Professor
Department of CSE
SSN College of Engineering
miltonrs@ssn.edu.in

Mirnalinee T T
Professor and Head
Department of CSE
SSN College of Engineering
mirnalineett@ssn.edu.in

Abstract

Our paper explores a task involving the analysis of emotions and triggers within dialogues. We annotate each utterance with an emotion and identify triggers, focusing on binary labeling. We emphasize clear guidelines for replicability and conduct thorough analyses, including multiple system runs and experiments to highlight effective techniques. By simplifying the complexities and detailing clear methodologies, our study contributes to advancing emotion analysis and trigger identification within dialogue systems.

1 Introduction

Emotion recognition and trigger detection in conversational data represent critical frontiers in natural language processing (NLP) research, offering profound insights into human-computer interaction, sentiment analysis, and dialogue understanding. In today's interconnected world, where communication transcends linguistic boundaries, understanding the subtle nuances of emotions expressed in code-mixed dialogues becomes increasingly imperative. Code-mixing, characterized by the seamless integration of multiple languages within a single conversation, reflects the rich tapestry of multicultural societies and presents unique challenges and opportunities for computational linguistics. Additionally, in monolingual English dialogues, identifying triggers—key points where emotional shifts occur—serves as a gateway to unraveling the underlying sentiment dynamics and contextual flow of conversations.

1.1 Significance of the Tasks

Our primary task, focuses on emotion recognition in code-mixed dialogues, holds immense significance in deciphering the intricacies of Code-Mixed communication. Accurately discerning emotions such as joy, sadness, anger, and more across diverse linguistic contexts enriches our understanding of cross-cultural expression and human sentiment. Meanwhile, the following tasks extend this exploration to trigger detection within both code-mixed and English dialogues. Identifying triggers not only facilitates the detection of emotional transitions but also provides deeper insights into the contextual triggers and socio-cultural factors shaping conversational dynamics.

1.2 Challenges and Opportunities

The complexity of code-mixed dialogues lies in disentangling the interplay of languages, cultural nuances, and emotional expressions. Herein lies the challenge of accurately recognizing emotions amidst linguistic diversity and cultural variations. Similarly, trigger detection in both code-mixed and English dialogues demands robust models capable of capturing subtle emotional shifts amid the fluidity of conversation. Addressing these challenges presents opportunities to develop sophisticated NLP techniques that transcend linguistic barriers and capture the essence of human emotions in their full complexity.

In our exploration, we experimented with Convolutional Neural Networks (CNN) Suseelan et al. (2019) and BERT models Sivanaiah et al. (2020) to tackle these challenges. However, we encountered some limitations. The CNN model yielded a low weighted F1 score of 0.28, indicating its struggle to effectively capture the nuances of emotional expression in code-mixed dialogues. On the other hand, while BERT showed promise in its ability to understand complex language patterns, it proved to

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 730–736

be computationally intensive, ultimately crashing after extended periods of runtime.

These setbacks highlight the need for further research and development in the field of NLP, particularly in the context of code-mixed dialogues and emotional recognition. Future efforts could explore novel model architectures, optimization techniques, and data augmentation strategies to improve performance and efficiency in emotion recognition and trigger detection tasks within code-mixed conversations. By addressing these challenges, we can pave the way for more accurate and reliable NLP solutions that better reflect the intricacies of human communication across diverse linguistic and cultural landscapes.

2 Overview

2.1 Summary of the task

The task involves recognizing emotions and detecting triggers in conversational data, with a focus on both Code-Mixed and English dialogues. Emotion recognition is structured as a classification task where systems predict the emotions associated with each utterance in a dialogue. Trigger detection entails identifying points in the conversation where emotional shifts occur. The datasets used include MaSaC for Hindi-English dialogues and MELD for English dialogues. The input comprises utterances from dialogues, and the output consists of predicted emotions for each utterance in emotion recognition, while trigger detection, indicates the presence or absence of triggers at each point in the dialogue.

2.2 Impact of the task

This task addresses the critical need for natural language processing (NLP) systems to understand and interpret emotions in conversational data. By focusing on code-mixed dialogues, it highlights the challenges posed by linguistic diversity and cultural nuances in emotion recognition. Additionally, the task emphasizes the importance of trigger detection in understanding the dynamics of conversations and capturing shifts in emotional states. By participating in this task, researchers contribute to advancing the capabilities of NLP systems in recognizing and understanding emotions in diverse linguistic contexts, thereby paving the way for more nuanced and culturally sensitive human-machine interactions.

3 Related Work

Emotion recognition and trigger detection in conversational data have been subjects of active research in natural language processing (NLP) and affective computing. Researchers have explored various approaches and methodologies to tackle these tasks, aiming to understand human emotions expressed in dialogue interactions and detect key points where emotional shifts occur. In this section, we review existing literature, highlighting recent advancements and key findings in the field.

The paper "Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed Text" introduces a novel approach to sentiment analysis in code-mixed social media data. They present a Hi-En code-mixed dataset and propose a Subword-LSTM architecture, enabling the model to capture sentiment information from important morphemes. This linguistic-driven approach outperforms traditional methods, achieving a notable accuracy improvement of 4-5% and surpassing existing systems by 18% in sentiment analysis of Hi-En code-mixed text Joshi et al. (2016).

Advancements in sentiment analysis techniques now recognize the temporal variability of emotions in textual data. For example, the SSN MLRG1 team at SemEval-2017 Task 4 introduced a novel approach using the Gaussian Process with fixed rule multi-kernel learning for sentiment analysis of tweets. Their method effectively captures evolving emotions by considering properties such as smoothness and periodicity. This approach aligns with our exploration of emotion recognition and trigger detection in conversational data, emphasizing the importance of incorporating temporal dynamics into sentiment analysis frameworks.S et al. (2017a)

This team also participated in task 5, focusing on fine-grained sentiment analysis. Their system utilizes Multiple Kernel Gaussian Processes to identify optimistic and pessimistic sentiments associated with companies and stocks. Given that comments on the same entities can exhibit varying emotions over time, considering properties like smoothness and periodicity becomes crucial. Their experiments highlight the effectiveness of the Multiple Kernel Gaussian Process in capturing diverse properties compared to a single Kernel Gaussian Process. S et al. (2017b)

In summary, existing research in emotion recognition and trigger detection in conversational data has explored diverse methodologies, including

deep learning, machine learning, rule-based approaches, and multimodal fusion techniques. Recent advancements have demonstrated the potential of context-aware features, multimodal data integration, and hybrid models in improving accuracy and robustness in these tasks. However, challenges such as linguistic diversity, cultural nuances, and ambiguity in emotional expressions continue to pose significant obstacles, warranting further research and exploration in the field.

4 Task Description

The tasks encompass emotion recognition and trigger detection in conversational data, each assessing specific competencies related to understanding emotions and detecting emotional shifts in dialogues.Kumar et al. (2024a)

4.1 Task 1 (ERC for code-mixed)

Objective: This task aims to evaluate the system's capability to recognize emotions in code-mixed dialogues, where multiple languages are used interchangeably.

Description: We were provided with a dataset consisting of code-mixed dialogues, where utterances contain a mix of languages. The task involves identifying the emotions expressed in each utterance accurately. Emotions may include disgust, contempt, anger, neutral, joy, sadness, fear, and surprise. Kumar et al. (2023)

4.2 Task 2 (EFR for code-mixed)

Objective: This task focuses on assessing the system's performance in detecting triggers that indicate emotional shifts in code-mixed dialogues.

Description: We were presented with code-mixed dialogues where emotional shifts occur. The task involves detecting these triggers within the dialogues. Triggers are specific instances or phrases that signal a change in the emotional tone of the conversation.Kumar et al. (2022)

4.3 Task 3 (EFR for English)

Objective: Similar to Task 2, this task evaluates the system's ability to detect triggers indicating emotional shifts. However, the focus is on English-only dialogues.

Description: We were provided with a dataset containing English-only dialogues. The task remains the same as Task 2, requiring us to identify triggers that signify emotional shifts within the conversations.Kumar et al. (2024b)

These tasks aim to assess the robustness and effectiveness of systems in understanding and interpreting emotional nuances within conversational data, particularly in Code-Mixed and English-only settings.

5 Experimental Setup

In this section, we provide a detailed overview of the experimental setup. Refer to 1 for the detailed architecture diagram illustrating the entire process.

5.1 Data Splits

The dataset provided for each task was split into three main subsets: training, development (dev), and testing. The distribution of data among these subsets was as follows:

5.1.1 Training Set

The training set comprised approximately 80% of the total dataset. This sizable portion allowed the models to learn patterns and associations from a diverse range of examples. It contained code-mixed and English-only dialogues with corresponding emotion labels for Task 1 and trigger labels for Tasks 2 and 3.

5.1.2 Development Set

The dev set accounted for around 10% of the dataset. It was utilized for fine-tuning the models' hyperparameters, such as regularization strength and feature extraction settings. This subset enabled us to iteratively adjust the model configurations to improve performance without overfitting to the training data.

5.1.3 Test Set

The test set constituted the remaining 10% of the dataset and was kept completely separate from the training and dev sets. It served as an unseen dataset for the final evaluation of model performance. Its purpose was to assess how well the trained models generalized to new, unseen instances and to provide an unbiased estimate of their performance.

5.2 Preprocessing

Before feeding the data into the machine learning models, we applied several preprocessing steps to ensure consistency and improve model performance:

5.2.1 Text Preprocessing

In our text preprocessing pipeline, we employed several steps to prepare the textual data for model

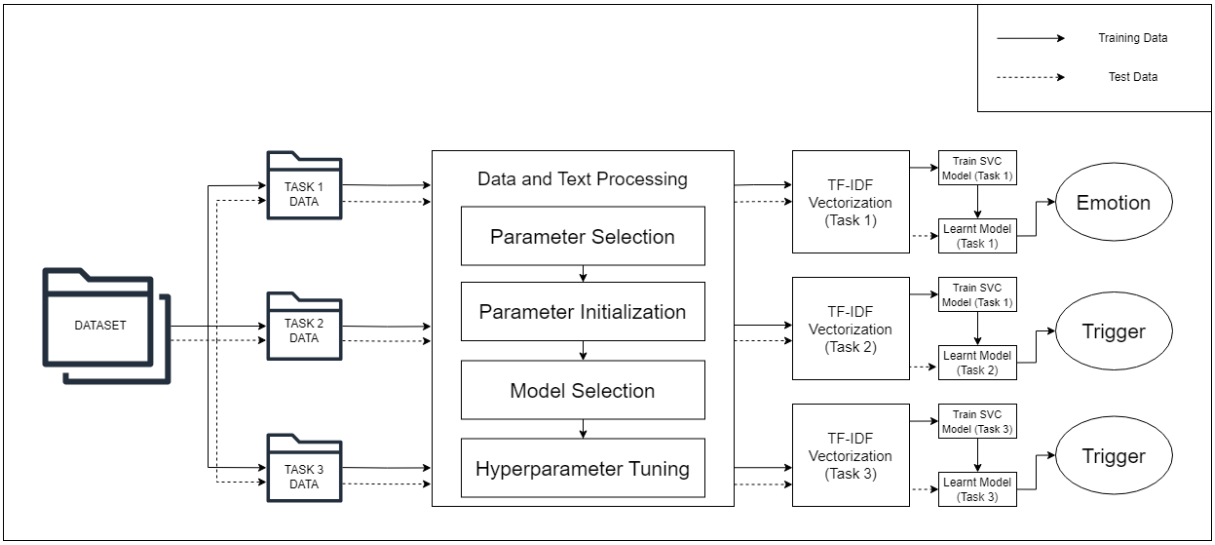


Figure 1: Architecture Diagram illustrating the entire Experimental Setup.

input. First, we tokenized the text using the word_tokenize function from the Garg and Sharma (2020)nltk library, splitting it into individual tokens or words. Next, we removed common stopwords, such as articles, prepositions, and conjunctions, using the predefined stopwords list provided by the nltk library, which helped reduce noise in the data. Additionally, we eliminated punctuation marks from the text to standardize the input and prevent the models from treating punctuation as meaningful features. This step involved removing characters such as periods, commas, and quotation marks. Finally, to ensure uniformity and improve generalization, we converted all text to lowercase, preventing the models from treating words with different cases as distinct features and effectively reducing the dimensionality of the input space.

5.3 Hyperparameter Tuning

Hyperparameter tuning was a crucial aspect of our experimental setup, as it involved optimizing the model's configuration to achieve the best performance on the dev set. We experimented with various hyperparameters, including:

5.3.1 TF-IDF Vectorization Parameters

We explored different settings for the Zhang et al. (2011)TF-IDF vectorization process, such as the $ngram\_range$ parameter, which determined the range of n-grams (contiguous sequences of words) considered during feature extraction. By adjusting the $ngram\_range$, we aimed to capture different combinations of words and phrases to better represent the text.

5.3.2 LinearSVC Parameters

For the Kaibi et al. (2019)LinearSVC classifier, we tuned parameters such as the regularization strength (C) to control overfitting. We also adjusted the *random_state* parameter to ensure reproducibility of results across different runs.

5.3.3 Grid Search with Cross-Validation

To find the optimal combination of hyperparameters, we employed grid search with cross-validation on the dev set Priyadarshini and Cotton (2021). This technique involved exhaustively searching through a specified parameter grid and evaluating each combination using cross-validation to estimate performance.

5.4 External Tools/Libraries Used

Our experimental setup relied on several external tools and libraries to facilitate data processing, model training, and evaluation. We leveraged scikit-learn for implementing various algorithms, data preprocessing tasks, and evaluation metrics. Additionally, the nltk library played a crucial role in performing natural language processing tasks such as tokenization, stopwords removal Mangat et al. (2017), and stemming Rao et al. (2021). We utilized joblib for saving and loading trained models to disk, providing a convenient way to serialize Python objects, including machine learning models. Pandas, a popular data manipulation library in Python, was instrumental in handling and analyzing structured data, enabling us to perform exploratory data analysis and prepare the data for training and evaluation. These external tools and

libraries streamlined our experimental workflow, allowing us to focus on model development and performance optimization.

6 Experimental Workflow

6.1 Task 1 (ERC for Code-Mixed dataset)

Data Preprocessing:

- Load the provided dataset containing code-mixed dialogues and their corresponding emotion labels.
- Perform text preprocessing steps such as tokenization, lowercasing, and removing stop words and punctuation.

Model Training:

- Utilize the preprocessed data to train a machine learning model, such as Linear Support Vector Classifier (LinearSVC), using the training set.
- Use techniques like TF-IDF Vectorization to convert text data into numerical features.

Evaluation:

- Evaluate the trained model's performance using the development set to fine-tune hyperparameters and ensure robustness.
- Evaluate the final model on the test set to measure its ability to accurately predict emotions in code-mixed dialogues.

6.2 Tasks 2 & 3 (EFR for Code-Mixed and English dataset)

Data Preprocessing:

- Load the provided dataset containing code-mixed or English-only dialogues and their corresponding trigger labels.
- Perform text preprocessing steps similar to Task 1.

Model Training:

- Train a machine learning model, such as LinearSVC, using the training set.
- Utilize techniques like TF-IDF Vectorization to convert text data into numerical features.

Evaluation:

- Evaluate the trained model's performance using the development set to fine-tune hyperparameters and ensure robustness.
- Evaluate the final model on the test set to measure its ability to accurately detect triggers indicating emotional shifts in dialogues.

7 Results

7.1 Evaluation

The model's performance is evaluated using accuracy (Acc), precision(P), recall(R), and F1 - Score (F1). These metrics are calculated as follows:

$$P = \frac{TruePositives}{TruePositives + FalsePositives} \quad (1)$$

$$R = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (2)$$

$$F1 = \frac{2 \times P \times R}{P + R} \quad (3)$$

7.2 Task 1 (ERC for code-mixed)

Main Quantitative Findings: The system achieved an accuracy of 48% on the test set. While the recall for the 'neutral' emotion is relatively high (81%), the precision and recall for other emotions are considerably lower, indicating challenges in accurately predicting emotions in code-mixed dialogues.

Quantitative Analysis: Table 1 presents the precision, recall, and F1-score for each emotion category. The results indicate that the model performed relatively well in identifying 'neutral' emotions but struggled with other emotions, particularly 'disgust' and 'sadness'.

Error Analysis: The model seems to have difficulties distinguishing between 'disgust' and 'contempt', as evidenced by low precision and recall for both categories. Further investigation is needed to understand the underlying causes of misclassifications.

7.3 Task 2 (EFR for code-mixed)

Main Quantitative Findings: The system achieved high precision, recall, and F1-score across all emotion categories in detecting triggers indicating emotional shifts in code-mixed dialogues.

Quantitative Analysis: Table 2 presents the precision, recall, and F1-score for each emotion category. The model performed exceptionally well

Table 1: Results for Task 1

Emotion	Precision	Recall	F1-Score
Disgust	0.00	0.00	0.00
Anger	0.34	0.14	0.20
Contempt	0.29	0.09	0.14
Neutral	0.52	0.81	0.63
Joy	0.45	0.29	0.35
Sadness	0.33	0.16	0.22
Fear	0.33	0.13	0.19
Surprise	0.42	0.24	0.31

Table 2: Results for Task 2

Emotion	Precision	Recall	F1-Score
Disgust	0.99	0.92	0.96
Anger	0.98	0.95	0.96
Contempt	0.97	0.94	0.95
Neutral	0.94	0.98	0.96
Joy	0.97	0.93	0.95
Sadness	0.96	0.93	0.95
Fear	0.98	0.92	0.95
Surprise	0.93	0.80	0.86

in identifying triggers for emotions such as anger, contempt, and fear. However, there was a slight decrease in recall for surprise, indicating some challenges in capturing subtle cues for this emotion category.

Error Analysis: The model demonstrated robust performance overall, with minor discrepancies in recall for certain emotion categories. Further investigation is warranted to understand the underlying causes of these discrepancies and refine the model's performance.

7.4 Task 3 (EFR for English)

Main Quantitative Findings: The system exhibited robust performance in detecting triggers indicating emotional shifts in English-only dialogues, achieving high precision, recall, and F1-score across all emotion categories.

Quantitative Analysis: Table 3 presents the precision, recall, and F1-score for each emotion category. The model demonstrated excellent precision and recall for most emotion categories. However, there was a slight decrease in recall for surprise, suggesting challenges in accurately capturing triggers for this particular emotion.

Error Analysis: Similar to Task 2, the model showcased strong overall performance, with minor discrepancies in recall for certain emotion categories. Further investigation is needed to address these discrepancies and enhance the model's accuracy in detecting emotional triggers.

Table 3: Results for Task 3

Emotion	Precision	Recall	F1-Score
Disgust	0.92	0.87	0.90
Anger	0.92	0.84	0.88
Contempt	0.92	0.97	0.94
Neutral	0.82	0.97	0.88
Joy	0.93	0.77	0.84
Sadness	0.92	0.84	0.88
Fear	0.95	0.80	0.87
Surprise	0.92	0.71	0.80

8 Conclusion

The exploration of emotion recognition and trigger detection in conversational data presents significant implications for natural language processing research and human-computer interaction. Our study, encompassing tasks focused on code-mixed dialogues and English-only conversations, sheds light on the challenges and opportunities inherent in understanding the nuanced expressions of human emotions.

Through our experimental endeavors, we have demonstrated the efficacy of machine learning models, particularly Linear Support Vector Classifier (LinearSVC), in recognizing emotions and detecting triggers within dialogues. Despite the complexities posed by linguistic diversity and cultural nuances, our systems have shown promising performance, especially in identifying triggers indicating emotional shifts.

However, our journey does not end here. Future work should delve deeper into understanding the underlying causes of misclassifications and explore innovative approaches to enhance model robustness and generalization. Additionally, incorporating context-aware features and leveraging advanced deep learning architectures could further improve the accuracy and granularity of emotion analysis in conversational data.

In conclusion, our study contributes to the advancement of emotion analysis and trigger detection within dialogue systems, paving the way for more nuanced and culturally sensitive human-machine interactions in diverse linguistic contexts. As we continue to unravel the intricacies of human emotions through computational linguistics,

we embark on a journey toward more empathetic and intuitive artificial intelligence systems.

References

Neha Garg and K Sharma. 2020. Annotated corpus creation for sentiment analysis in code-mixed hindi-english (hinglish) social network data. *Indian Journal of Science and Technology*, 13(40):4216–4224.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491.

Ibrahim Kaibi, Hassan Satori, et al. 2019. A comparative evaluation of word embeddings techniques for twitter sentiment analysis. In *2019 International conference on wireless technologies, embedded and intelligent systems (WITS)*, pages 1–4. IEEE.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024a. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2024b. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, 5(3):1339–1348.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Veenu Mangat et al. 2017. Dictionary based sentiment analysis of hinglish text. *International Journal of Advanced Research in Computer Science*, 8(5).

Ishaani Priyadarshini and Chase Cotton. 2021. A novel lstm–cnn–grid search-based deep neural network for sentiment analysis. *The Journal of Supercomputing*, 77(12):13911–13932.

Himanshu Singh Rao, Jagdish Chandra Menaria, and Satyendra Singh Chouhan. 2021. A novel approach for sentiment analysis of hinglish text. In *Mathematical Modeling, Computational Intelligence Techniques and Renewable Energy: Proceedings of the First International Conference, MMCITRE 2020*, pages 229–240. Springer.

Angel Deborah S, S Milton Rajendram, and T T Mirnalinee. 2017a. SSN_MLRG1 at SemEval-2017 task 4: Sentiment analysis in Twitter using multi-kernel Gaussian process classifier. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 709–712, Vancouver, Canada. Association for Computational Linguistics.

Angel Deborah S, S Milton Rajendram, and T T Mirnalinee. 2017b. SSN_MLRG1 at SemEval-2017 task 5: Fine-grained sentiment analysis using multiple kernel Gaussian process regression model. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 823–826, Vancouver, Canada. Association for Computational Linguistics.

Rajalakshmi Sivanaiah, Angel Suseelan, S Milton Rajendram, and Mirnalinee T.t. 2020. TECHSSN at SemEval-2020 task 12: Offensive language detection using BERT embeddings. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2190–2196, Barcelona (online). International Committee for Computational Linguistics.

Angel Suseelan, S Rajalakshmi, B Logesh, S Harshini, B Geetika, S Dyaneswaran, S Milton Rajendram, and TT Mirnalinee. 2019. Techssn at semeval-2019 task 6: Identifying and categorizing offensive language in tweets using deep neural networks. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 753–758.

Wen Zhang, Taketoshi Yoshida, and Xijin Tang. 2011. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert systems with applications*, 38(3):2758–2765.

TüDuo at SemEval-2024 Task 2: Flan-T5 and Data Augmentation for Biomedical NLI

Veronika Smilga and **Hazem Alabiad**
University of Tübingen, Germany
first.last@student.uni-tuebingen.de

Abstract

This paper explores using data augmentation with smaller language models under 3 billion parameters for the SemEval-2024 Task 2 on Biomedical Natural Language Inference for Clinical Trials. We fine-tune models from the Flan-T5 family with and without using augmented data automatically generated by GPT-3.5-Turbo and find that data augmentation through techniques like synonym replacement, syntactic changes, adding random facts, and meaning reversion improves model faithfulness (ability to change predictions for semantically different inputs) and consistency (ability to give same predictions for semantic preserving changes). However, data augmentation tends to decrease performance on the original dataset distribution, as measured by F1 score. Our best system is the Flan-T5 XL model fine-tuned on the original training data combined with over 6,000 augmented examples. The system ranks in the top 10 for all three metrics[1].

1 Introduction

In the recent years, the rapid and triumphant advance of Large Language Models (LLMs) has affected virtually every area of NLP, biomedical NLP included. We aim to prove that Biomedical NLP can still benefit from smaller models of no more than three billion parameters. First, as the saying goes, "You must not use a steam hammer to crack a nut, if a nutcracker would do". In other words, while LLMs' performance is unmatched in complex applications, smaller models may be perfectly sufficient for simpler tasks, such as text classification or natural language inference. Second, being pre-trained on extremely large corpora of unlabelled data, modern LLMs have been shown to exhibit dataset-related bias (Acerbi and Stubbersfield, 2023). In fields with a high error cost, pre-training and fine-tuning models on smaller, carefully curated, high-quality datasets is safer and more predictable than using black-box giant LLMs in a zero-shot or few-shot setting. Finally, as of now, best-performing state-of-the-art LLMs are either largely unavailable to the end-user due to computational constraints (for open-source models) or cost-inefficient (for proprietary models with access via API).

The NLI4CT-2024 Shared Task (Jullien et al., 2024) consists in building a system for natural language inference (NLI) based on a collection of breast cancer Clinical Trial Reports (CTRs) in English. The task's main challenge is the complex and heterogeneous nature of the data. For each datapoint, the premise comes from one of the four sections of a CTR – Intervention, Eligibility, Results, or Adverse Events. Naturally, the sections are different from each other in terms of the mean length, the proportion of numerical data present, and the level of world knowledge required for drawing conclusions. Compared to the previous year's iteration of the task (Jullien et al., 2023), this year's challenge calls for a system robust to alterations in the data. Apart from F1 measure, two new metrics are used to evaluate the model performance: **faithfulness**, "measuring the ability of a model to correctly change its predictions when exposed to a semantic-altering intervention", and **consistency**, "measuring the ability of a system to predict the same label for original statements and contrast statements for semantic preserving interventions".

According to the last year participants' reports, various augmentation techniques have not led to significant performance improvement in terms of F1 and the top-3 best-performing systems did not use data augmentation at all (Jullien et al., 2023). However, given the new metrics that are used in this year's evaluation, it seems reasonable to continue exploring the effect that various kinds of augmentation have on F1, faithfulness, and consistency at the same time. In this paper, we fine-tune mod-

[1]Our code is available at https://github.com/smilni/semeval2024_safe_biomedical_nli

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 737–744

els of Flan-T5 family with and without the use of augmented data automatically generated using GPT-3.5-Turbo. We find that using various kinds of additional data leads to an increase in model's faithfulness and consistency, but a decrease in F1. Our best system is Flan-T5 XL, fine-tuned on 1900 original train and development instances and 6650 automatically generated ones. The system ranks 7th for consistency, 9th for faithfulness, and 10th for F1.

2 Related work

Language Models and Biomedical NLP There has been a surge of LLMs fine-tuned on biomedical data, from relatively small – 7 billion parameter ChatDoctor (Yunxiang et al., 2023), MedAlpaca (Han et al., 2023), PMC-LLAMA (Wu et al., 2023); 6 billion parameter DoctorGLM (Xiong et al., 2023) and OphGLM (Gao et al., 2023) – to extremely large ones – 540B Med-PaLM (Singhal et al., 2023), 175B Codex-Med (Liévin et al., 2022), 80B Med-Flamingo (Moor et al., 2023) – which were reported to break state-of-the-art results on a number of biomedical NLP tasks. Moreover, without any fine-tuning on biomedical data, GPT-4 was reported to have passed every step of the US-medical licensing exam (Nori et al., 2023). However, researchers argue that smaller language models, such as T5 Base and T5 Large, still outperform gigantic all-purpose models when fine-tuned for a specific task (Lehman et al., 2023).

Model Robustness in NLI tasks Many NLI models suffer from bias related to superficial correlations between input text features and labels in the training dataset, which leads to a drop in performance on datasets where these correlations do not hold (Rajaee et al., 2022). Among those are hypothesis only bias, where models rely mostly on the hypothesis without taking premise and premise-hypothesis relations into account (Poliak et al., 2018), and word-overlap bias, where models rely on the presence of shared words or phrases in premise and hypothesis (McCoy et al., 2019). Various techniques may be used to mitigate this bias, such as adversarial training (Stacey et al., 2020) and data augmentation with predicate-argument structures (Moosavi et al., 2020) and syntactic transformations (Min et al., 2020).

3 Experimental setup

In our experiments, we aim to test whether language models of relatively small size, under three billion parameters, can achieve decent performance on a task with a simple objective – as the model chooses between only two options, entailment and contradiction – and complex data – as dealing with Clinical Test Reports requires complex reasoning and understanding of numerical data. As a starting point for the experiments, we have chosen Flan-T5.

3.1 Selecting the model

Flan-T5 (Chung et al., 2022) is an updated checkpoint of T5 (Raffel et al., 2020), instruction-fine-tuned on a number of new NLP tasks, which outperforms baseline T5 models of the corresponding size on a number of benchmarks. It also features improved instruction-following capabilities and generalizes well on new tasks, not present in the training data. On the previous year's iteration of NLI4CTR, the system that featured fine-tuned Flan-T5-XXL (Kanakarajan and Sankarasubbu, 2023) without any biomedical pre-training data augmentation showed an impressive performance, ranking second.

First, we evaluate the model's performance in three scenarios – zero-shot, few-shot and after fine-tuning. Due to computational constraints we limit our experiments to language models of under three billion parameters, so we test only Flan-T5 Small (80M parameters), Flan-T5 Base (250M parameters), Flan-T5 Large (780M parameters), and Flan-T5 XL (3B parameters).

When testing the models in a zero-shot setting, we and use one of the NLI prompt templates provided by Flan-T5 developers [2]. In cases where two CTRs are given, we concatenate them using newline character as a separator. For a few-shot setting, we use the same prompt template as on the previous step, but enhance it with two hand-picked short CTR-hypothesis pairs from the training set – one with entailment and the other with contradiction relation. Refer to Appendix A for the prompts used for querying Flan-T5 in a zero-shot and two-shot setting.

Finally, we carry out fine-tuning with the use of HuggingFace Transformers library on the entire train set. The same set of hyperparameters is used for all models: `auto_find_batch_size`

[2] https://github.com/google-research/FLAN/blob/main/flan/v2/flan_templates_branched.py

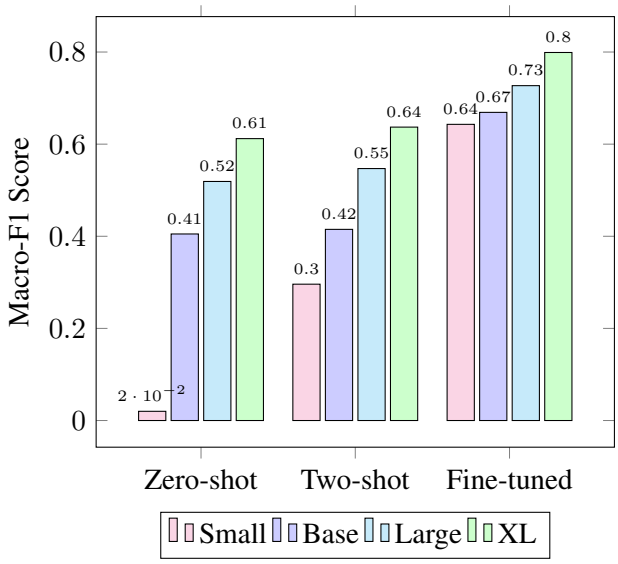


Figure 1: Flan-T5 model family performance, calculated on development dataset

`= True`, `learning_rate = 1e-3`, `optimizer = adamw_torch`. Flan-T5 Small and Flan-T5 Base are fine-tuned for 5 epochs in full precision. Due to computational constraints, Flan-T5 Large and Flan-T5 XL, however, were fine-tuned for 3 epochs using int8 precision and HuggingFace implementation of Low-Rank Adaptors (LoRA) algorithm (Hu et al., 2021).

Figure 1 summarizes the results obtained after evaluating the model on the development set in three different settings: zero-shot, few-shot, and after fine-tuning. There is a clear correlation between the model's performance and its size and between the model's performance and the number of train examples provided to it as well. In all cases, providing two examples from the training data to the model in a few-shot setting improves the performance of the model slightly, while fine-tuning it on all given training data results in a substantial performance boost. The best-performing model so far is fine-tuned Flan-T5 XL.

3.2 Data augmentation

We assume that the training data should be augmented in two key ways to create a **faithful** and **consistent** system. First, we should add paraphrased versions of the original datapoints, with semantic meaning and label preserved. It will ensure that the system is **consistent**, i.e. produces the same output for semantically equivalent inputs. Second, we should include semantically altered versions of the original datapoints, with semantic meaning changed and a reverted label assigned. It will ensure that the system is **faithful**, i.e. change its output when encountering an input semantically different to the one seen before. The presence of these three types of datapoints – original, paraphrased in a semantically preserving way, and paraphrased in a semantically altering way – is expected to improve the model's performance. We assume that these examples will teach the model to consistently handle the semantics of the sentence, mitigating the impact of superficial features like word overlap between a premise and a hypothesis on the model's performance.

We apply four types of alterations to hypotheses:

1. Synonym-based semantic-preserving changes, where certain words within a sentence are substituted with their synonymous counterparts.

2. Syntactic semantic-preserving changes, where the syntactic structure of the sentence is changed while the semantic meaning remains the same.

3. Random fact addition semantic-preserving changes, where a true random fact is appended to the hypothesis without affecting its truth value.

4. Semantic-altering changes, where a sentence contradictory to the original hypothesis is formulated.

Semantic-preserving changes 1), 2), and 3) are applied to all hypotheses, while semantic-altering change 4) is only applied to hypotheses that were

Original hypothesis	Heart-related adverse events were recorded in both the primary trial and the secondary trial. `[entailment]`
Synonym-based alteration	*Cardiovascular* adverse events were *documented* in both the primary *study* and the secondary *study*. `[entailment]`
Syntactic alteration	Both the primary trial and the secondary trial recorded adverse events related to the heart. `[entailment]`
Random fact addition	*Lymphadenopathy is the enlargement of lymph nodes due to infection, inflammation, or cancer.* Heart-related adverse events were recorded in both the primary trial and the secondary trial. `[entailment]`
Semantic-altering change	Heart-related adverse events were *not* recorded in either the primary trial or the secondary trial. `[contradiction]`

Table 1: Examples for each kind of alterations

initially labeled as entailment, changing the label to contradiction. The reason for this decision is that reverting a hypothesis that follows from some text produces a hypothesis that contradicts this text, but not vice versa. You may find examples for each kind of alterations in Table 1.

We access GPT-3.5-Turbo via OpenAI API to generate new hypotheses for each CTR-hypothesis pair, using a distinct hand-crafted prompt for each kind of alteration. You may find the text of each prompt in Appendix B. Four new hypotheses are generated for each "entailment" CTR-hypothesis pair, with both semantic-preserving and semantic-altering changes applied, and three new hypotheses are generated for each "contradiction" CTR-hypothesis pair, with only semantic-preserving changes applied. In all cases, CTR text itself remains unaltered, and only hypothesis is affected.

As a result, we obtain 3400 new entries for 850 original train CTR-hypothesis pairs labelled as entailment and 2550 new entries for 850 original train CTR-hypothesis pairs labelled as contradiction. The process of generating 5950 data points, thus increasing our dataset by 4.5 times, cost $0.86 and took 1.5 hours to complete.

4 Results

4.1 Individual Augmentation Analysis

We fine-tune Flan-T5 XL model on augmented data using the same set of hyperparameters as in Section 3.1. First, we fine-tune the model separately on each type of augmented data (combined with the original data) to estimate how augmentation of each kind affects the performance. The results are presented in Table 2.

Interestingly, only one kind of augmentation, the synonym-based one, had a positive effect on the model's performance on the original dataset, while the others led to a decrease in F1. All kinds of augmentations resulted in a model with higher consistency, i.e. a model better at producing the same output for hypotheses with the same meaning. The alteration that consisted in adding random true facts to hypotheses led to the highest increase in consistency. However, only semantic-altering change resulted in a more faithful model, i.e. a model better at changing its prediction when encountering a similar but semantically different hypothesis. All semantic-preserving changes led to a decrease in the model's faithfulness. Overall, our data augmentation techniques have proven to be efficient in improving the model's robustness. However, they have simultaneously resulted in a worse performance on the original data.

4.2 Final Model Selection

The next step was to try out different combinations of augmented data to reach the optimal performance in terms of the largest increase in both faithfulness and consistency and the smallest decrease in terms of F1. As the goal of the competition was to create a faithful and consistent system, we prioritized these metrics over F1 when choosing the model for the final submission. Thus, we chose the model trained on the entire set of augmented data that demonstrates higher faithfulness and consistency but lower F1. For the final submission, we additionally enriched the dataset with 200 more entries from development data and 700 new augmented entries created using techniques described in Section 3.2. The results obtained after fine-tuning the model on the entire augmented dataset are presented in Table 3.

	F1	Faithfulness	Consistency
Original train data only	0.779	0.780	0.667
Original train data + synonym-based alterations	**0.780**	0.715	0.681
Original train data + syntax-based alterations	0.764	0.748	0.698
Original train data + random facts addition	0.748	0.736	**0.725**
Original train data + reverted meaning alterations	0.735	**0.854**	0.686

Table 2: Flan-T5 XL performance when trained on different kinds of augmented data, calculated on test dataset

	F1	Faithfulness	Consistency
Original train data only	**0.779**	0.780	0.667
Original train data + all augmented train data	0.745	**0.851**	0.748
Original train and dev data + all augmented train and dev data	0.760	0.841	**0.752**

Table 3: Flan-T5 XL performance when trained on all kinds of augmented data, calculated on test dataset

4.3 Other approaches

Numerical inference is a known challenge for large language models. We assumed that the model's performance might vary across different CTR sections, with a decrease in performance for sections that contain most numbers. To check this assumption, we calculated the final model's F1 for each section separately. Calculations were performed on the development dataset as we had no access to test dataset labels during the development and evaluation stages. The results are presented in Table 4.

	F1
Adverse Events	0.711
Eligibility	0.821
Intervention	0.861
Results	0.759
All sections	0.783

Table 4: Final model's performance on each CTR section, calculated on development dataset

Adverse events, the section that, according to our observations, most often contained numbers in premise as well as hypothesis and required numerical inference to determine the relation between them, had the lowest F1 of all.

We attempted to develop a separate model, Flan-T5 XL with the same hyperparameter set as in Section 3.1, to tackle CTR-hypothesis pairs of this kind. The model was first pre-fine-tuned on EQUATE dataset (Ravichander et al., 2019) for 3 epochs in an attempt to enhance its numerical inference capabilities. Then it was further fine-tuned on the original and augmented CTR-hypothesis pairs of Adverse Events category for 3 epochs as well. We then used the original model to produce predictions for Eligibility, Intervention, and Results sections and the new model to produce predictions for Adverse Events section. However, on test data, this approach resulted in a decrease in performance with an F1 of 0.756 (-0.004), faithfulness of 0.781 (-0.06) and consistency of 0.722 (-0.031). We suppose that the decrease in performance is explained by the fact that the second model, trained on ~1/4 of all data (only one section out of four), simply did not encounter enough data to develop robustness comparable to that of the final model trained on the entire dataset.

5 Conclusion

In this paper, we explore the impact of data augmentation on model performance and robustness. Specifically, we focus on leveraging advanced language models like GPT-3.5-Turbo to expand the training set for fine-tuning smaller models such as Flan-T5 XL. Our experiments involve various prompts to generate new CTR-hypothesis pairs. Enriching the training set with new examples that underwent semantic-preserving changes, such as synonym replacement, change in word or clause

order, and random true fact addition, improves the model's consistency. Adding augmented examples that underwent semantic-altering changes, such as meaning reversion, improves the model's faithfulness as well as consistency. However, all kinds of augmentation except for synonym replacement lead to a decrease in model performance in terms of F1 on the original unaltered dataset. The model selected for the final submission is Flan-T5 XL fine-tuned on augmented development and training set. It features higher robustness but lower base performance than Flan-T5 XL fine-tuned on original data only, with faithfulness of 0.841 (+0.061), consistency of 0.752 (+0.085), and F1 of 0.76 (-0.019).

Acknowledgements

We would like to thank Dr. Çağrı Çöltekin for guiding, supporting and inspiring us on our way.

References

Alberto Acerbi and Joseph M Stubbersfield. 2023. Large language models show human-like content biases in transmission chain experiments. *Proceedings of the National Academy of Sciences*, 120(44):e2313790120.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Weihao Gao, Zhuo Deng, Zhiyuan Niu, Fuju Rong, Chucheng Chen, Zheng Gong, Wenze Zhang, Daimin Xiao, Fang Li, Zhenjie Cao, et al. 2023. Ophglm: Training an ophthalmology large language-and-vision assistant based on instructions and dialogue. *arXiv preprint arXiv:2306.12174*.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Tianyu Han, Lisa C Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno K Bressem. 2023. Medalpaca–an open-source collection of medical conversational ai models and training data. *arXiv preprint arXiv:2304.08247*.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Donal Landers, and André Freitas. 2023. Semeval-2023 task 7: Multi-evidence natural language inference for clinical trial data. *arXiv preprint arXiv:2305.02993*.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama ai research at semeval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003.

Eric Lehman, Evan Hernandez, Diwakar Mahajan, Jonas Wulff, Micah J Smith, Zachary Ziegler, Daniel Nadler, Peter Szolovits, Alistair Johnson, and Emily Alsentzer. 2023. Do we still need clinical language models? *arXiv preprint arXiv:2302.08091*.

Valentin Liévin, Christoffer Egeberg Hother, and Ole Winther. 2022. Can large language models reason about medical questions? *arXiv preprint arXiv:2207.08143*.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. *arXiv preprint arXiv:2004.11999*.

Michael Moor, Qian Huang, Shirley Wu, Michihiro Yasunaga, Yash Dalmia, Jure Leskovec, Cyril Zakka, Eduardo Pontes Reis, and Pranav Rajpurkar. 2023. Med-flamingo: a multimodal medical few-shot learner. In *Machine Learning for Health (ML4H)*, pages 353–367. PMLR.

Nafise Sadat Moosavi, Marcel de Boer, Prasetya Ajie Utama, and Iryna Gurevych. 2020. Improving robustness by augmenting training sentences with predicate-argument structures. *arXiv preprint arXiv:2010.12510*.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Sara Rajaee, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. Looking at the overlooked: An analysis on the word-overlap bias in natural language inference. *arXiv preprint arXiv:2211.03862*.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. Equate: A benchmark evaluation framework for quantitative reasoning in natural language inference. *arXiv preprint arXiv:1901.03735*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Sebastian Riedel, and Tim Rocktäschel. 2020. Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training. *arXiv preprint arXiv:2004.07790*.

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Further finetuning llama on medical papers. *arXiv preprint arXiv:2304.14454*.

Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Qian Wang, and Dinggang Shen. 2023. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*.

Li Yunxiang, Li Zihan, Zhang Kai, Dan Ruilong, and Zhang You. 2023. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv preprint arXiv:2303.14070*.

A Prompts used in querying Flan-T5

```
Read the text and determine if the sentence
is true:
{premise}
Sentence: {hypothesis}
["yes", "no"]
```

Figure 2: Zero-shot prompt used in querying Flan-T5.

```
Read the text and determine if the sentence
is true:
Inclusion Criteria: Estrogen receptor or
progesterone receptor positive breast cancer
Premenopausal with regular menstrual cycles
Exclusion Criteria: Current oral
contraceptives
Sentence:  Males are not eligible for the
primary trial.
["yes", "no"]
Answer: yes

(another instruction, CTR, and answer - this
time with a contradiction relation)

Read the text and determine if the sentence
is true:
{premise}
Sentence: {hypothesis}
["yes", "no"]
Answer:
```

Figure 3: Two-shot prompt used in querying Flan-T5.

B Prompts used to obtain augmented data from GPT-3.5-Turbo

```
TEXT: {text}
Paraphrase TEXT using synonyms while
preserving its original meaning. Always keep
words "primary trial" and "secondary trial"
if present in TEXT, do not replace "primary
trial" and "secondary trial" with synonyms.
Return only paraphrased text and nothing
else.
```

Figure 4: Prompt used to generate a synonym-based paraphrased version of hypothesis.

```
TEXT: {text}
Change the syntactic structure of TEXT while
preserving its original meaning. Always keep
words "primary trial" and "secondary trial"
if present in TEXT, do not replace "primary
trial" and "secondary trial" with synonyms.
Return only paraphrased text and nothing
else.
```

Figure 5: Prompt used to generate a syntax-based paraphrased version of hypothesis.

```
Generate a random short true definition of a
random medical term. Use format '{term} is
{definition}'. Definition must be no longer
than one sentence."
```

Figure 6: Prompt used to generate a random biomedical fact to then append to hypothesis.

```
TEXT: {text}
Revert the original meaning of TEXT. The
result must contradict TEXT. Return only the
result and nothing else.
```

Figure 7: Prompt used to generate sentence with meaning contradicting that of hypothesis.

FeedForward at SemEval-2024 Task 10: Trigger and sentext-height enriched emotion analysis in multi-party conversations

Zuhair Hasan Shaik[1], **R Dhivya Prasanna**[1], **Enduri Jahnavi**[1], **Rishi Koushik Reddy Thippireddy**[1], **P S S Vamsi Madhav**[1], **Sunil Saumya**[1], and **Shankar Biradar**[1]
[1]Department of Data Science and Intelligent Systems,
Indian Institute of Information Technology Dharwad, Dharwad, Karnatka, India
(zuhashaik12, prasanna0083,jahnavienduri,rishikoushik18,pssmvamsi)
@gmail.com (sunil.saumya, shankar)@iiitdwd.ac.in

Abstract

This paper reports on an innovative approach to Emotion Recognition in Conversation and Emotion Flip Reasoning for the SemEval-2024 competition with a specific focus on analyzing Hindi-English code-mixed language. By integrating Large Language Models (LLMs) with Instruction-based Fine-tuning and Quantized Low-Rank Adaptation (QLoRA), this study introduces innovative techniques like Sentext-height and advanced prompting strategies to navigate the intricacies of emotional analysis in code-mixed conversational data. The results of the proposed work effectively demonstrate its ability to overcome label bias and the complexities of code-mixed languages. Our team achieved ranks of 5, 3, and 3 in tasks 1, 2, and 3 respectively. This study contributes valuable insights and methods for enhancing emotion recognition models, underscoring the importance of continuous research in this field.

1 Introduction

Emotional analysis has come quite a long way. In the context of natural language processing (NLP), history reveals an evolution of the emotion analysis task. The task has always been about recognizing emotions from text, evolving from those early-day systems that were able to recognize emotions from standalone text (Akhtar et al., 2019; Chatterjee et al., 2019; Mageed and Ungar, 2017; Shankar Biradar and Chauhan, 2021) to the current cutting-edge challenge of Emotion Recognition in Conversation (ERC) (Lei et al., 2023; Hazarika et al., 2018). Well-designed simple methods have demonstrated that recognizing the emotion of a user's expression enables a broad range of practical applications in diverse domains, from e-commerce (Gupta et al., 2013) to healthcare (Khanpour and Caragea, 2018).

ERC plays a significant role in illustrating how the emotion change during the interpersonal communications. By contrast to the isolation of single texts, ERC struggles with how emotions shift through a combination of different speakers in conversation. Motivated by the urgent need to understand the complex interactions of emotions during dialogue, a new issue has arisen—Emotion-Flip Reasoning (EFR) (Kumar et al., 2022a, 2024b). EFR is a novel Endeavour aiming at identifying precisely which utterances transform an emotion within a person's flow of speech. Apart from just emotions, EFR seeks to unravel the complexities of emotion flips, offering valuable insights into the dynamics of human interaction. Emotional flips can result from internal party interactions or from external elements such as speaker gestures or verbal messages.

The practical importance of EFR extends beyond theoretical limitations. In reality, it has applications in a variety of sectors. EFR plays a crucial part in the development of reward and punishment systems, as well as interpretable emotion recognition systems. Further, the widespread use of Hindi-English code-mixed language online shows the cultural change. NLP is facing new challenges in the accurate identification of emotions in a dynamic cultural context. Language switching during the conversation makes the work of emotion recognition systems even more complex. Further building an adaptable system capable of capturing the subtle variations in emotions that emerge in such a hybrid language setting is the need of the hour.

In order to promote research in this field, the organisers of SemEval 2024, Emotion Discovery and Reasoning its Flip in Conversation (EDiReF) [1] organised a shared task. The organisers created three sub-tasks:

- Task 1: Emotion Recognition in Conversation (ERC) in Hindi-English code-mixed conversations

[1]https://lcs2.in/SemEval2024-EDiReF/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 745–756

- Task 2: Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations
- Task 3: EFR in English conversations.

The following is an illustration of the definition for the ERC and EFR tasks:

- Emotion Recognition in Conversation (ERC) is focused on assigning emotions to individual utterances or phrases within a dialogue. It involves analyzing conversation data to identify the emotional states expressed by speakers throughout the interaction. The goal of ERC is to accurately recognize and categorize the emotions conveyed in each utterance.
- Emotion Flip Reasoning (EFR) aims to identify triggers for emotion flips in multi-party conversations. A trigger can be caused by one or more utterances, and some emotion flips might not be triggered by other speakers but by the target utterance itself (self-trigger). EFR analyzes dialogue data to understand the causes behind shifts in emotions, providing insights into the dynamics of emotional exchanges in conversations.

Our team, FeedForward, participated in Semeval-2024 task 10 and achieved rankings of 5, 3, and 3 in subtasks 1, 2, and 3, respectively[2]. For detailed insights and findings regarding this task, please refer to the task description paper of SemEval-2024 Task 10 (Kumar et al., 2024a).To tackle this problem, we propose state-of-the-art techniques such as Sentext-height for emotion recognition in multi-party conversations and ratio-wise splitting in trigger datasets for the EFR task. Additionally, we utilized instruction-based QLoRA training of 7-billion-parameter models for both ERC and EFR tasks.

The outline of the article is as follows: Section 2 offers an in-depth exploration of the background study. In addition, Section 4 comprehensively discusses the proposed methodologies. Finally, the experimental outcomes are illustrated in section 5.

2 Related work

Emotion detection in the standalone text is a well-known challenge in the Natural Language Processing domain (Akhtar et al., 2019; Chatterjee et al., 2019). However, unlike single text, emotion recognition in conversation data requires numerous complicated understandings of contextual information and speakers (Wagh and Sutar, 2023). In accordance with this, the majority of studies used deep neural networks with memory functions to solve sophisticated understandings of conversational text data (Hazarika et al., 2018; Weston et al., 2014). Furthermore, the developers of (Zhong et al., 2019) attempt to include the role of speakers into the conversational model by using memory networks during two-party discussions.

The utilization of external information is also vital in recognizing emotions in multi-party conversations. The authors of (Wen et al., 2023) proposed the DIMMN network for capturing speaker interaction information during multi-party conversations, in addition to text, audio, and video aspects during experiments. Conventional categorical label-based approaches fail to capture quantitative measurements of emotion; to solve this issue, the authors of (Yang et al., 2023) created a low-dimensional cluster-level contrastive learning model incorporating linguistic and factual information. Furthermore, the (Li et al., 2023) established a discourse link between utterances by adding symbolic information into multi-party interactions.

ERC in low-resource code-mixed text has received little attention. The authors of (Ghosh et al., 2023; Saumya et al., 2022) created a Hindi-English emotion-annotated corpus and established a transformer-based end-to-end framework with multitask learning. Furthermore, most existing studies only account for emotion recognition, but very few studies looked beyond emotion recognition to interpret the results. In one such study, (Kumar et al., 2022b; Fharook et al., 2022), the authors introduced a novel Emotion-Flip Reasoning (EFR), which aims to identify past utterances that have triggered one's emotional state to flip at a certain time, in addition to ERC.

3 Dataset

3.1 MaSac_ERC

The organizers of EDiReF of SemEval 2024 have provided the MaSac_ERC dataset (Kumar et al., 2023) for emotion recognition in Hindi-English Code-Mixed Conversations (Task 1). The task is to recognize emotions for speaker utterances in conversations. The train dataset contains 343 conversations and a total of 8506 utterances, which

[2]All proposed models are openly available at: https://huggingface.co/collections/zuhashaik/multi-party-dialoz-65d34c9f74e0888ef4e66da3

contain 8 emotion classes—Neutral, Joy, Anger, Sadness, Contempt, Fear, Surprise, and Disgust. The data set is significantly skewed, and the distribution of emotions across the train, validation, and test data is shown in Figure 1.

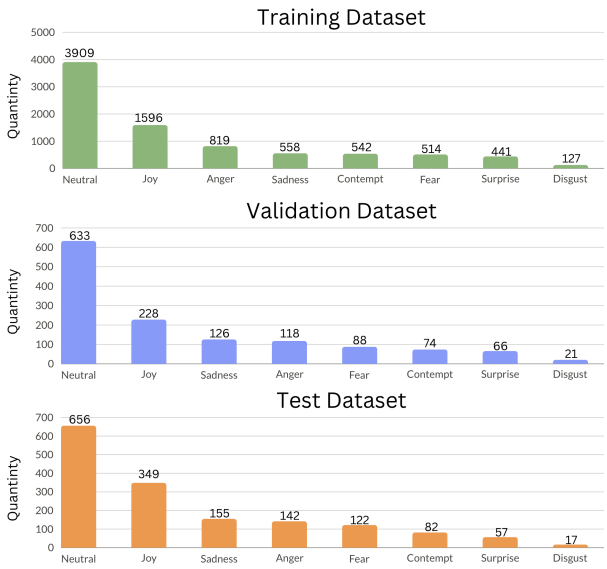


Figure 1: Emotion distribution of the MaSac_ERC

3.2 MaSac_EFR and MELD_EFR

The organizers of EDiReF of SemEval 2024 have provided the MaSac_EFR and MELD_EFR datasets Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations (Task 2) and English Conversations (Task 3) respectively. The goal is to find all utterances that trigger a flip in the emotion of a speaker within a conversation. The MaSac_EFR train dataset contains 4,893 conversations having 6,542 triggers and 92,233 non-triggers. And the dataset distribution is clearly illustrated in Table 1. Similarly the MELD_EFR dataset contains 4,000 conversations having 5,575 triggers and 29,425 non-triggers. And the data distribution of triggers and non-triggers is illustrated in Table 2.

Trigger	Train	Validation	Test
Yes (1)	6542	434	416
No (0)	92233	7024	7274

Table 1: MaSac_EFR Label distribution

Trigger	Train	Validation	Test
Yes (1)	5575	494	1169
No (0)	29425	3028	7473

Table 2: MELD_EFR Label distribution

4 Methodology

In this section, a comprehensive study of the methodology employed, focusing on Emotion Recognition in Conversations (ERC) in Hindi-English code-mixed data and Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations, as well as in English for Task1, Task2, and Task3, respectively, for SemEval-2024 Shared Task-10.

4.1 Task 1 : ERC in Hinglish

In this study, the focus lies on examining emotions within Hindi-English (Hinglish) code-mixed multi-party conversations using advanced language models. Various methods are explored, including refining BERT derivatives and translating code-mixed utterances for emotion classification. Furthermore, strategies like simplifying emotion labels and utilizing large language models with effective prompts are implemented to improve performance.

4.1.1 BERT derivatives as Baseline

As is commonly known, BERT (Devlin et al., 2019) demonstrates exceptional proficiency in sentiment analysis across various domains in natural language processing (NLP). However, the dataset comprises Hindi-English code-mixed text, necessitating pre-trained BERT derivatives capable of understanding Hinglish.

After an extensive exploration and experimentation phase with various BERT models, several BERT derivatives trained on Hindi or Hindi-English code-mixed datasets were identified. These include bert-base-multilingual-cased[3]**??**, l3cube-pune's hing-mbert-mixed-v2 (Joshi, 2023), lxyuan's distilbert-base-multilingual-cased-sentiments-student, and papluca's xlm-roberta-base-language-detection. Additionally, google's FNet-base (Lee-Thorp et al., 2022) was considered due to its substantial research presence in sentiment analysis, showcasing promising outcomes.

In this approach, each utterance paired with its corresponding emotion was treated as a data point extracted from the MaSac_ERC dataset. Subsequently, this data was utilized to fine-tune BERT derivatives for the emotion classification task, irrespective of its position within the conversation sequence and relevant contextual nuances.

[3]https://huggingface.co/google-bert/bert-base-multilingual-cased

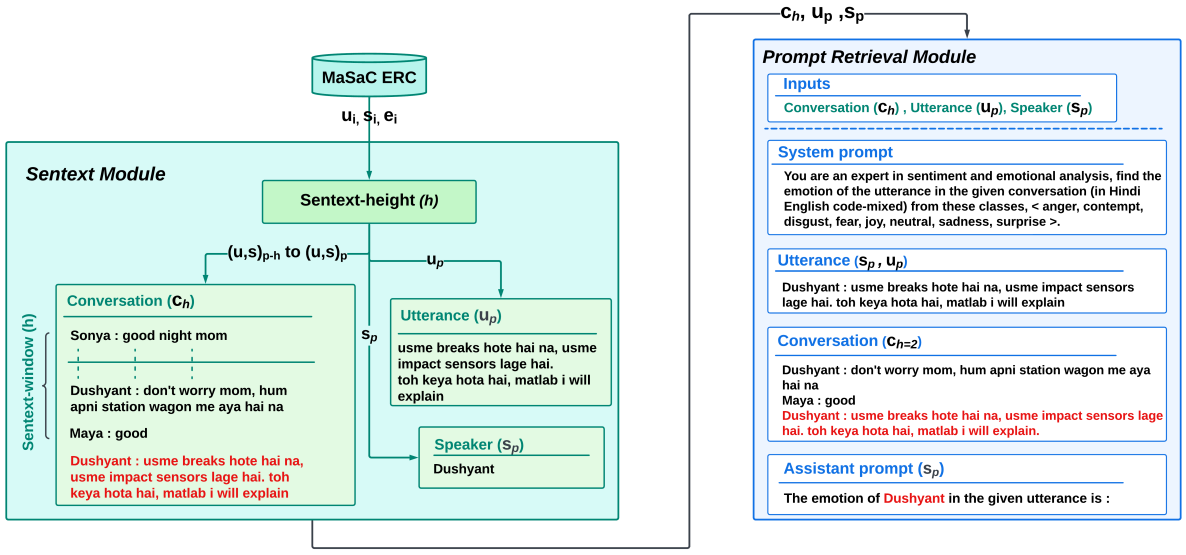


Figure 2: In the overview of the SP Module, the figure illustrates the complete process from slicing conversations with the Sentext Module to obtaining a training-ready prompt from the Prompt Retrieval Module.

In this approach, all layers of the models were retained unfrozen, converging into a feedforward network and subsequently a dense 8-way classifier, empowered by softmax.

4.1.2 Hinglish to English Translation

In the study focusing on Emotion Recognition in Code-Mixed Hindi-English Conversations (ERC), a unique methodology was employed. Rather than following a sequential conversation analysis, the code-mixed utterances were transliterated and then translated using IndicXlit (Madhani et al., 2023) and IndicTrans2 (Gala et al., 2023), respectively from AI4BHARAT organization. The inference of the models and the procedure of converting Hinglish to English are accessible here.[4] The translated utterance with its corresponding emotion was then used as a data point to fine-tune BERT and FNet for the sequence classification task.

4.1.3 Split and concat

In the split and concat approach, the label was coarse-grained (Neutrals, Negatives, Positives) to study the nuances created by the labels and the dataset complexity. Then, Fine grained to only Negatives (Anger, Sadness, Contempt, Fear, and Disgust) and only Positives (Joy, Surprise) were considered.

The main aim of this approach is to create a ensemble architecture (a classifier's tree) that will reduce the complexity of the dataset for the models being used. At the first level, it classify sentences as Neutral, Negative, or Positive. Then, at the second level, it further classify negatives and positives.

For instance, at the first level, An *NNP* (Neu-Neg-Pos) classifier predicts the sentiment of the utterance as Neutral, Negative, or Positive. If it's Neutral, the process stops there as we already classifed the emotion. Otherwise, it proceeds to the corresponding output sentiment classifier (Negs or Pos) to further classify the fine grained emotion.

4.1.4 7Bs enhanced with SP-module

When traditional approaches failed to yield satisfactory results, primarily due to label bias and the complexity of the Hindi-English code-mixed language, which struggled to distinguish between classes effectively, the focus shifted to large language models. 7-Billion (7B) parameter Large Language Models (LLMs) were utilized, taking these models from the shelf and then finetuning using Quantized Low Rank Adaptation QLoRA (Dettmers et al., 2023) on the dataset with effective prompts.

Sentext-height

To enhance the model's performance, a novel concept called *Sentext-height* was introduced. Sentext-height is a new idea that comes from context related to sentiment analysis within a sentence. It determines how many previous

[4]The proposed methodology can be found here: https://github.com/Zuhashaik/Multi-Party-DialoZ

utterance influence the emotion analysis of the present utterance in a given conversation. With this, it is possible to capture the emotion state of a speaker in the past utterances, which can contribute to finding the emotion of the present speaker's utterance.

Prompt-engineerning

LLMs have proven to be significantly reliable for a wide array of tasks in the domain of NLP. While they show significant promise, effective usage requires a carefully curated input. Through extensive experimentation with prompt structures on the foundational models, a conclusion was reached with a prompt that effectively works for the model.

The structure of the Prompt Retrieval Module:

- System prompt: Defines the LLMs role and expected behavior within the interaction, guiding its response.
  <|*system*|>*You are an expert in sentiment and emotional analysis, find the emotion of the utterance in the given conversation (in Hindi-English code mixed) from these classes, [anger, contempt, disgust, fear, joy, neutral, sadness, surprise].*

- Utterance: This contains the present utterance (u_p) with the respective speaker (s_p) attached to it before the utterance.
  <|*utterance*|> *{Speaker}:{Present_utterance}*

- Conversation: This has the conversation that is driven by sentext-height (h). It consists of h+1 utterances with Sentext-window (u_{p-h} to u_{p-1}) along with the current utterance u_p which to be evaluated, each with their corresponding speakers identified to indicate who made those utterances.
  <|*conversation*|> *{conversation, h}*

- Assistant prompt: Provides an incomplete statement or scenario and expects LLM to finish the very next word, making it a classification task that we're interested in.
  <|*assistant*|>*The emotion of {Speaker (s_p)} in the given utterance is :*

  In this case, the probable choices are the various emotions listed in the system prompt. These emotions include anger, contempt, disgust, fear, joy, neutral, sadness, and surprise. The model tries to classify within these emotion categories.

Data preprocessing hence concludes with the setting the sentext-height and selection of the appropriate prompt, collectively referred to as the SP-module (Sentext-Prompt) and clearly illustrated in the Figure 2.

QLoRA and Instruction Finetuning

After preparing the data with the SP-module, we used the prompt-processed dataset to fine-tune 7Bs with Instruction-based QLoRA for classifying emotions. We made 6 datasets, altering the sentext-height (h) from 2 to 7. Each model will train on every dataset, and we'll choose the best sentext-height based on how well the model performs. The models employed in this proposed study include Llama-2-7b-chat-hf (Touvron et al., 2023), zephyr-7b-beta (Tunstall et al., 2023), Mistral-7B-Instruct-v0.1 (Jiang et al., 2023), and openchat_3.5 (Wang et al., 2023).

Due to the challenges posed by *Catastrophic forgetting* (Luo et al., 2023) and computational constraints, the full training of LLMs (7Bs) cannot be carried out. Instead, QLoRA was chosen. This method involves quantizing the model during inference and then applying LoRA. With LoRA, the model parameters are frozen, and an additional low-rank matrix is introduced beside the attention layer weights, rather than training all parameters. This approach significantly reduces training time and memory requirements, often resulting in improved performance compared to traditional fine-tuning methods.

Additionally, a custom classifier was designed, where the last decoder layer in the 7B LLM is connected to an 8-way dense network powered by a softmax classifier. This is distinct from the text-generation LLM, where the 7B LLM is connected to a vocab-sized (32,000 in this case) classifier to predict the next word of the given input, which iterates until the end of sequence tag <eos> arises or the token limit is reached.

The total integration of the Sentext-height, Prompt-module and custom architecture with LoRA are demostrated in the figure 3.

Experimental setup

In this case, all models are inferred and trained in FP16 (Half-precision, float16). Following extensive experiments with various sentext-height

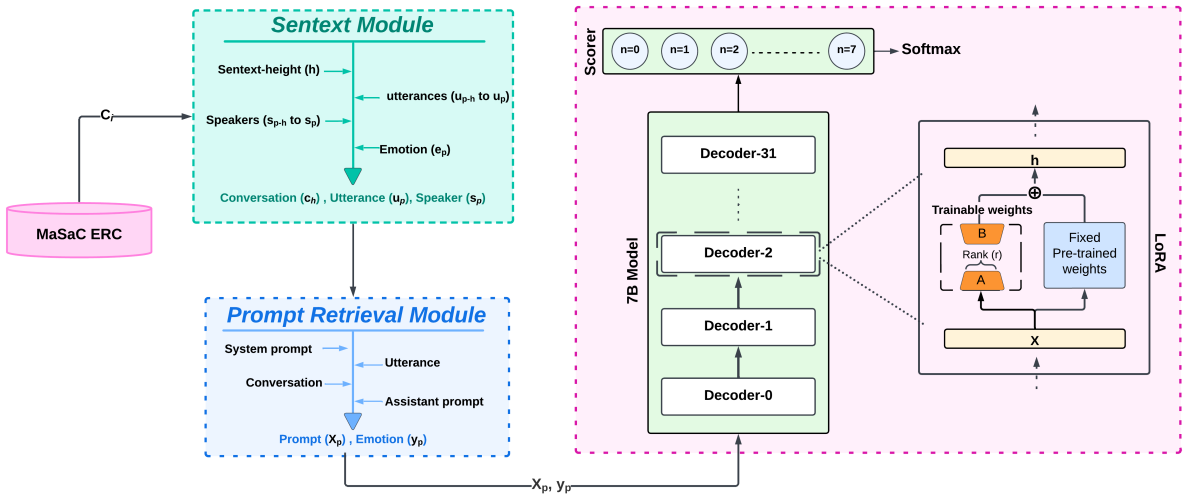


Figure 3: The MaSac-ERC-Z framework, figure displays how the Sentext Module and Prompt Retrieval Module are combined with a 7-billion parameter LLM. It also shows how LoRA is incorporated into the model, with each decoder having a low-rank matrix next to the pre-trained attention weights. This LoRA technique is applied specifically to all 32 decoder layers.

(h={2-7}), the hyperparameters that proved effective for the proposed model have been identified, as outlined in Table 3. Considerable

Hyper parameter	Value
Rank (LoRA config)	16
LoRA Alpha (LoRA config)	64
Dropout (LoRA config)	0.2
Learning Rate	2×10^{-5}
Learning Rate Scheduler	Constant
Batch size	1
Gradient acumulation step	1
adam_beta1	0.9
adam_beta2	0.999
adam_epsilon	1.000×10^{-8}
rms_norm_eps	1.000×10^{-5}

Table 3: Hyper parameters for Training 7Bs

RAM and computing capabilities are leveraged, supported by 3×32G Nvidia Tesla V100 GPUs.

4.2 Task 2 and 3 : EFR in Hinglish and English respectively

Since both task 2 and 3 involve Emotion Flip Reasoning but in different languages, maintaining the core model while adjusting the input is proposed. When providing embeddings to the model, rich semantic information from the text in the same language as the dataset is ensured. This approach enables obtaining language-aware contextual embeddings for the core model under development.

4.2.1 Attention-Based Utterance Fusion

In this approach, the Bert-based embeddings $(e_1, e_2, ..e_n)$ are extracted for each and every utterance $(u_1, u_2, ..u_n)$ in the conversation of n utterances. Consider u_p as the present utterance from the conversation, and the task is to determine whether it is the trigger for the u_n utterance which led to an emotion flip. Now, u_p and u_n are considered, and their embeddings e_p and e_n respectively are obtained. These embeddings are then linearly concatenated and passed through multi-head attention to capture intricate patterns within concatenated utterance pairs. Subsequently, a feed-forward network followed by a binary classifier is applied. Experimentation has been conducted with different BERT derivatives and the number of heads in multi-head attention has been varied.

4.2.2 7Bs for EFR

Following the MaSac-ERC-Z framework used in Task 1 with the 7B language model, the similar architecture is adopted here. However, a 2-way dense softmax classifier is incorporated instead of 8, as the task aims for binary classification (trigger or non-trigger). Furthermore, the focus is solely on identifying triggers rather than analyzing conversational emotion, so the sentext module is omitted. Additionally, a specialized prompt module is introduced to enhance the efficiency of

trigger retrieval for the specific task.

Prompt Module
After extensive experimentation in the playground of the foundational models, a prompt that works effectively for this task was concluded.
The glance of the prompt:

- System prompt: Defines the LLMs role and expected behavior within the interaction, guiding its response.
  <|*system*|>*In your role as an expert in sentiment and emotion analysis, your primary objective is to identify trigger utterances for emotion-flips in multi-party conversations (in Hindi-English code-mixed). Evaluate the provided dialogue by analyzing changes in emotions expressed by speakers through their utterances. Your task is to determine the accuracy of the hypothesis based on these emotional shifts.*
  For Task 3, which is the MELD dataset in English, *(in Hindi-English code-mixed)* from the system prompt is removed, and the remaining architecture will remain the same.

- Hypothesis: This contains the hypothesis and expecting the LLM to evaluate the hypothesis.
  <|*Hypothesis*|> *The utterance <{present_utterance}> is a trigger for the emotion-flip in <{speaker}'s> : <{final_utterance}> in the conversation*

- Conversation: This section contains the entire conversation, ensuring no chance of missing context. The emotions are also provided immediately after each utterance in the conversation, which is crucial for identifying the emotion flip and analyzing which utterance is the trigger.
  <|*conversation*|> *{conversation}, {emotions}*

- Assistant prompt: A sentence is left incomplete, assuming that the LLM has already generated something related to the input task. The expectation is for the LLM to complete this sentence.
  <|*assistant*|> *The given Hypothesis is :*

Instruction and QLoRA finetuning
As discussed, this approach follows a similar method proposed in Task-1, the MaSac-ERC-Z module, where a dataset is created using the prompt module and Instruction-based QLoRA fine-tuning is performed for the 7B model on the dataset. However, the constraint is that there are 98,775 (6,542 Triggers and 92,233 Non-triggers) and 35,000 (5,575 Triggers and 29,425 Non-triggers) datapoints from the MaSac_EFR and MELD_EFR datasets respectively. This can significantly slow down the trainings and take a lot of time to complete. Experimentations would be impractical under these circumstances. To avoid these constraints, the dataset was sliced into an 1:n ratio of Triggers to Non-Triggers, where n = {1,2,..}.
For instance, in the Task 2 dataset (MaSac_EFR), there are 6,542 triggers and 92,233 non-triggers. To preserve all triggers, the same number of non-triggers was selected to create a 1:1 dataset, yielding 13,084 datapoints from a total of 98,775. Similarly, for a 1:2 ratio, 6,542 triggers were retained, and 13,084 non-triggers were selected, and so forth up to a 1:3 ratio. This reduction in dataset size resulted in shorter training times leading to more efficient model training.

5 Results

This section presents a comprehensive study on outcomes from Task 1, Task 2, and Task 3. The weighted-f1 score is used as the standard metric for all tasks, as recommended by the task organizers and utilized to evaluate the submission hosted on Codalab.

5.1 Task 1

The baseline
In the proposed study, BERT derivatives were utilized, among which mBERT exhibited significant performance, yielding a weighted F1 score of 41.70. Consequently, this served as an initial baseline for evaluating the effectiveness of subsequent ideas and models. The corresponding scores are provided in Table 4.

Base-Model	Weighted-F1
mBERT	**41.70**
hing-mbert-mixed-v2	28.76
lxyuan	40.25
papluca	37.39
fnet-base	38.08

Table 4: Weighted-F1 scores of Finetuned models

Translation
Following the initial efforts, the aim was to

enhance performance further, considering the intricate nature of deciphering patterns within code-mixed languages. The approach involved converting Hinglish (a mix of Hindi and English) into English and using transformer-based models to identify the emotions. After this transformation, a weighted-f1 score of 40.03 was achieved with bert-base-uncased and 35.79 with fnet-base. The decline is assumed to be the accumulation of errors across three key processes: transliteration, translation, and classification. These processes inherently carry a high risk of errors, which likely impacted the classification accuracy.

The classifier tree

In the proposed work, *Split and concat* in Task1, the impact of coarse and fine-grained approaches on classification was analyzed. This examination aimed to pinpoint areas for improvement in achieving scores above the baseline. The primary challenge lies in classifying Neutrals within the complex Hindi-English code-mixed context, resulting in a weighted-f1 score of 55.16. Additionally, categorizing fine-grained negatives poses a significant challenge, as evidenced by a weighted-f1 score of 39.87. However, identifying positives proves comparatively easier, with weighted-f1 of 91.28.

The strengths of all three classifiers were combined, resulting in an aggregate score of 41.46 with BERT-Tree. The Ensemble BERT-Tree consists of Hing-BERT as the first-level classifier (NNP) and mBERT for further classifying negatives (Negs) and positives (Pos) at the second level. These models were chosen based on their performance scores in both coarse and fine-grained classification tasks. Nonetheless, this represents a decline from the baseline as discussed earlier. The decrease may be due to the compounding errors from each classifier that affect the final classification.

The detailed investigation of the study is outlined in Table 5. In the table, "Neutral-Negative-Positive" represents the coarse grain classification, while "Negatives (Negs)" and "Positives (Pos)" indicate the fine grain emotion categories.

7Bs enhanced with SP-module

In the analysis, the proposed approaches fell short of delivering satisfactory results, preventing the achievement of a weighted-f1 score in the 50s. The complexity of code-mixed languages posed a significant challenge, and the methods struggled to grasp the nuances of context and sentiments effectively.

Base model	NNP	Negs	Pos
mBERT	49.42	**39.87**	**91.28**
hing-mbert	**55.16**	11.80	79.47
lxyuan	49.71	39.01	90.69
papluca	53.80	32.89	90.35
fnet-base	48.69	26.70	89.91
W-F1 (ALL)			
BERT-Tree (En)		**41.46**	

Table 5: Weighted-F1 scores of Coarse and Fine-Grained Emotion Classification Results, combined all to construct a tree like classifer to classify all 8 emotions. 'En' denotes Ensemble here.

However, upon transitioning to 7Bs for this task and conducting extensive experimentation on various foundational models and sentext-height (choosing n between 2-7), a threshold of 50s was finally surpassed, which elevated the system to the 5th position in the competition. Specifically, a weighted-f1 score of 51.17 was attained using the Zephyr-7b-beta model with a sentext-height of 3.

Based on the analysis presented, Table 6 and figure 4 illustrates the performance of various 7B models across different sentext-height (h) values.

7B Models	Sentext-height (h)					
	2	3	4	5	6	7
llama2	49.0	**49.5**	48.3	49.0	48.3	47.9
zephyr	46.7	**51.2**	45.5	46.0	47.4	46.3
mistral	45.5	45.5	44.5	**46.1**	45.5	47.0
openchat	42.4	46.7	47.3	45.0	43.2	**48.6**

Table 6: Weighted-F1 scores of 7B models with different Sentext-height (h) values.

All-Together

Bringing everything together for Task 1, the study began with mBERT as the benchmark, followed by efforts to refine performance through translation and ensemble techniques, which encountered challenges and resulted in reduced scores. Further exploration into classification strategies revealed difficulties in nuanced identification, leading to mixed outcomes. Finally, incorporating SP-modules helped to surpass the 50s score threshold, reflecting progress in addressing the complexities of code-mixed languages. The comprehensive results for Task 1 can be viewed in Table 7, providing a detailed overview of the study's findings.

Model	Model Names	W-F1	Method
Encoder-Only	mBERT	41.7	Seq-cls
	hing-mbert	28.76	Seq-cls
	lxyuan	40.25	Seq-cls
	papluca	37.39	Seq-cls
	fnet-base	38.08	Seq-cls
	BERT	40.03	Translation
	FNet	35.79	Translation
	BERT-Tree	41.46	Ensemble
Decoder-Only	llama_h3	49.52	QLoRA
	mistral_h3	45.5	QLoRA
	zephyr_h3	**51.17**	**QLoRA**
	openchat_h3	46.73	QLoRA
	mistral_h5	46.07	QLoRA
	openchat_h7	48.58	QLoRA

Table 7: The table provides weighted F1 scores comparison of various methods and models. For Decoder-only models, the sentext-height is specified after the model's name. "Seq-cls" denotes sequence classification.

5.1.1 Task 2 and Task 3

7Bs for EFR

In Task 1, 7Bs demonstrated remarkable performance, motivating the extension of their use to EFR. As outlined in the Methodology, training requires a significant amount of time due to the large number of data points. To address this, the dataset was sliced and implemented a 1:n ratio (Triggers : Non-Triggers), resulting in (1+n)x datapoints (where x represents the number of triggers).

This concept was applied to Task 3 as well, given the similar nature of Task 2 but with English-language data, ensuring consistency in the approach across both tasks.

Task2			
7B models	**1:1**	**1:2**	**1:3**
openchat	57.81	55.60	58.51
zephyr	66.19	66.32	76.96

Table 8: Task 2, Weighted-F1 scores of 7B models with different splitting ratios

Task3			
7B models	**1:1**	**1:2**	**1:3**
openchat	71.52	71.29	72.53
zephyr	70.77	71.97	71.91

Table 9: Task 3, Weighted-F1 scores of 7B models with different splitting ratios

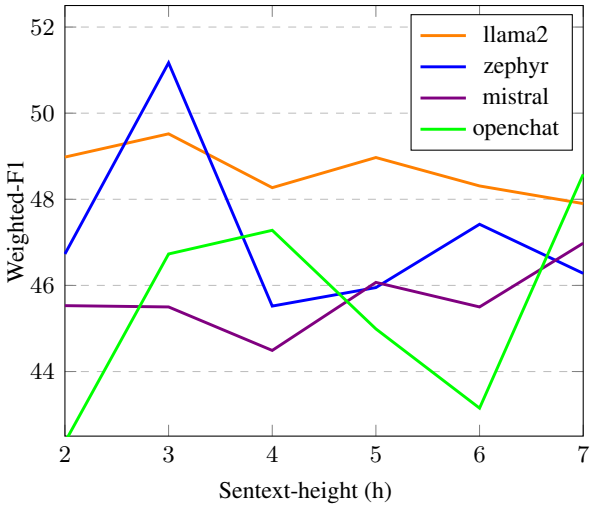


Figure 4: Graphical representation of the performance of 7B models with different Sentext-height (h) values.

For task 3, we considered the validation set and trained with a specific ratio (3:1) and model (openchat_3.5) that resulted in the highest score (72.53), achieving a weighted F1 score of 73.94 From the demonstrated experiments, the ratio of 1:3 yielded the highest scores in both Task 2 and Task 3, resulting in securing the 3rd rank in both tasks respectively. The results of the experiments with various ratio's and 7B models is given in the table 8 for MaSac_EFR which is task2 and table 9 for MELD_EFR which is task3.

6 Conclusion

This paper discusses the proposed work for the competition EDiReF SemEval-2024 hosted on Codalab. The study mainly focuses on emotion and emotion flip-trigger analysis specifically within multi-party conversational data. Through innovative approaches and the utilization of state-of-the-art techniques such as Large Language Models (LLMs), Instruction-based fine-tuning, and Quantized Low-Rank Adaptation (QLoRA), our team achieved promising results in Emotion Recognition in Conversation (ERC) and Emotion Flip Reasoning (EFR) tasks. However, obstacles persist, especially in addressing label bias and capturing nuanced emotions in Hindi-English code-mixed language. The findings underscore the need for further research to enhance model performance, ultimately improving emotional analysis in conversational data.

References

Md Shad Akhtar, Deepanway Ghosal, Asif Ekbal, Pushpak Bhattacharyya, and Sadao Kurohashi. 2019. All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework. *IEEE transactions on affective computing*, 13(1):285–297.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. Semeval-2019 task 3: Emocontext contextual emotion detection in text. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 39–48.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Shaik Fharook, Syed Sufyan Ahmed, Gurram Rithika, Sumith Sai Budde, Sunil Saumya, and Shankar Biradar. 2022. Are you a hero or a villain? a semantic role labelling approach for detecting harmful memes. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 19–23.

Jay Gala, Pranjal A. Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M. Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages.

Soumitra Ghosh, Amit Priyankar, Asif Ekbal, and Pushpak Bhattacharyya. 2023. Multitasking of sentiment detection and emotion recognition in code-mixed hinglish data. *Knowledge-Based Systems*, 260:110182.

Narendra Gupta, Mazin Gilbert, and Giuseppe Di Fabbrizio. 2013. Emotion detection in email customer care. *Computational Intelligence*, 29(3):489–505.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2018, page 2122. NIH Public Access.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Raviraj Joshi. 2023. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages.

Hamed Khanpour and Cornelia Caragea. 2018. Fine-grained emotion detection in health-related online posts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1160–1166.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024a. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2024b. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, 5(3):1339–1348.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022a. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022b. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2022. Fnet: Mixing tokens with fourier transforms.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework.

Wei Li, Luyao Zhu, Rui Mao, and Erik Cambria. 2023. Skier: A symbolic knowledge integrated model for conversational emotion recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul NC, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2023.

Aksharantar: Open indic-language transliteration datasets and models for the next billion users.

Muhammad Abdul Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 718–728.

Sunil Saumya, Vanshita Jha, and Shankar Biradar. 2022. Sentiment and homophobia detection on youtube using ensemble machine learning techniques. In *Working Notes of FIRE 2022-Forum for Information Retrieval Evaluation (Hybrid). CEUR*.

Sunil Saumya Shankar Biradar and Arun Chauhan. 2021. mbert based model for identification of offensive content in south indian languages. In *Working Notes of FIRE 2021-Forum for Information Retrieval Evaluation (Online). CEUR*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Nanda R Wagh and Sanjay R Sutar. 2023. Enhanced emotion recognition for women and children safety prediction using deep network. *International Journal of Intelligent Systems and Applications in Engineering*, 11(10s):500–515.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data.

Jintao Wen, Dazhi Jiang, Geng Tu, Cheng Liu, and Erik Cambria. 2023. Dynamic interactive multiview memory network for emotion recognition in conversation. *Information Fusion*, 91:123–133.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Kailai Yang, Tianlin Zhang, Hassan Alhuzali, and Sophia Ananiadou. 2023. Cluster-level contrastive learning for emotion recognition in conversations. *IEEE Transactions on Affective Computing*.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. *arXiv preprint arXiv:1909.10681*.

A Performance of Top model for Task 1

In the following appendix, we present the performance metrics of the instruction-tuned zephyr-7b-beta model (zephyr_h3) with Sentext-height (h=3) which is the top performing model with a Weighted-F1 of 51.17 in sub task 1, Emotion Recognition in Conversation (ERC).

A.1 Confusion Matrix

The confusion matrix in figure 5 visually represents the performance of the zephyr_h3 model in classifying different emotions. We observed a notable amount of confusion primarily between the emotions of joy and neutral, which could be attributed to the prevalence of neutral expressions in the dataset. This suggests a bias towards categorizing ambiguous or mild emotions as neutral, potentially impacting the accuracy of our predictions.

Additionally, there appears to be confusion between the emotions of anger and fear, as well as between contempt and sadness. These overlaps indicate potential similarities in the facial expressions or textual cues associated with these emotions, highlighting areas where our model may require further refinement.

A.2 Classification Report

The comprehensive classification report in table 10 for the zephyr_h3 model, showcasing precision, recall, F1 score, and support across various emotions.

The report further underscores the performance of our model across different emotions. While achieving relatively high precision for joy and neutral emotions, indicating a good ability to correctly identify these categories, our model struggles with emotions such as disgust and fear, as evidenced by lower precision scores. This indicates a tendency

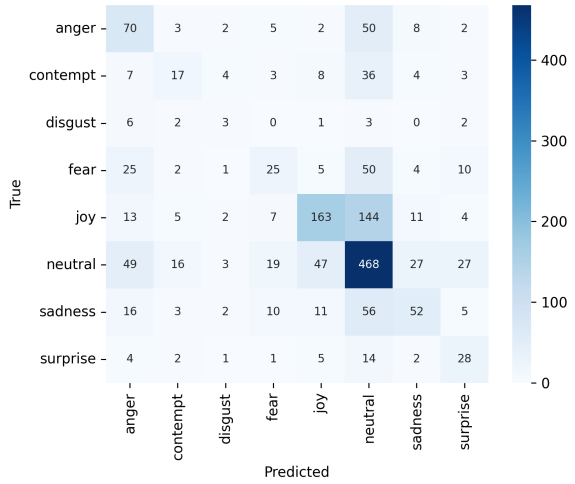


Figure 5: Confusion Matrix

for the model to misclassify instances of these emotions as other classes. Moreover, the overall accuracy of our model is moderate, indicating room for improvement in effectively distinguishing between the diverse emotional states. These findings emphasize the importance of addressing biases in the dataset and further fine-tuning the model to enhance its ability to accurately classify a wider range of emotions.

Emotion	**Precision**	**Recall**	**F1 Score**	**Support**
Anger	0.38	0.52	0.44	142
Contempt	0.33	0.20	0.25	82
Disgust	0.19	0.18	0.18	17
Fear	0.31	0.20	0.24	122
Joy	0.69	0.48	0.56	349
Neutral	0.58	0.72	0.65	656
Sadness	0.47	0.35	0.40	155
Surprise	0.34	0.47	0.39	57
Accuracy		0.53		
Macro Avg	0.41	0.39	0.39	1580
Weighted-Avg	0.53	0.53	0.52	1580

Table 10: Classification Report

A.3 Performance summary

Our classification model demonstrates moderate overall accuracy of 53%, with strengths in identifying joy and neutral emotions, boasting precision scores of 69% and 58% respectively. However, it struggles with emotions such as disgust and fear, showing lower precision scores of 19% and 31% respectively. Confusion primarily arises between joy and neutral emotions, possibly due to dataset biases towards neutral expressions. Further refinement is needed to enhance the model's ability to accurately classify a broader spectrum of emotions. The macro-average F1 score is 39%, while the weighted average F1 score is 52%, indicating room for improvement in capturing the nuances of different emotional states.

YNU-HPCC at SemEval-2024 Task 5: Regularized Legal-BERT for Legal Argument Reasoning Task in Civil Procedure

Peng Shi, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
Shipeng1@stu.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

This paper describes the submission of team YNU-HPCC to SemEval-2024 for Task 5: The Legal Argument Reasoning Task in Civil Procedure. The task asks candidates the topic, questions, and answers, classifying whether a given candidate's answer is correct (True) or incorrect (False). To make a sound judgment, we propose a system. This system is based on fine-tuning the Legal-BERT model that specializes in solving legal problems. Meanwhile, Regularized Dropout (R-Drop) and focal Loss were used in the model. R-Drop is used for data augmentation, and focal loss addresses data imbalances. Our system achieved relatively good results on the competition's official leaderboard. The code of this paper is available at https://github.com/YNU-PengShi/SemEval-2024-Task5.

1 Introduction

The task can be formulated as follows: given an introduction to the topic, a question, and an answer candidate, classify if the given candidate is correct (True) or incorrect (False) (Bongard et al., 2022). This task has two main difficulties: 1) The text length of the topic and question is much larger than 512 tokens. 2) The number of positive and negative samples in the data varies widely.

Initially, the online system represented the first attempt to utilize computational methods for addressing legal conundrums (VALENTE et al., 1999). Despite notable advancements in recent years, which have seen a concerted effort to establish objective benchmarks for natural language processing models in the domain of legal language comprehension (Chalkidis et al., 2022), a lack remains in the realm of complex tasks involving argumentative reasoning within legal contexts. However, Legal-BERT has emerged as a forerunner in this domain, demonstrating compelling performance (Chalkidis et al., 2020).

This paper proposes a model based on Legal-BERT. In processing tasks, we used sliding window simple (SWS) and sliding window complex (SWC) to process the original data and solved the problem of the token count of the original data being much larger than 512. In the subsequent process, we found that there was a significant imbalance in the dataset that resulted in the return of the most common label in the training set (in this case, 0). We added R-Drop (Wu et al., 2021) to the model to address this issue and changed the loss function from cross entropy to focal loss (Lin et al., 2017). In the end, we achieved a good result. The best submission for the test set has achieved 0.6166 and ranked 9th in this task.

The remainder of this paper is organized as follows. Section 2 describes the model and method used in our system, section 3 discusses the results of the experiments, and finally, the conclusions are drawn in section 4.

2 System Description

This section delves into the intricate design of the proposed model's architecture. The architecture comprises multiple essential components, namely the text cutting, the tokenizer, the pre-trained Legal-BERT model, the output layer, and the methods. Figure 1 illustrates the comprehensive system model that we have devised.

2.1 Text Preprocessing

Sliding Window Simple (SWS). The process involves dividing the combined question and introduction into discrete segments or chunks. These chunks are then submitted to a classification algorithm, which assigns a category or label to each segment based on its content. Once the classification is complete, the system calculates the average predicted output for all the chunks. This average serves as a comprehensive summary or representa-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 757–762

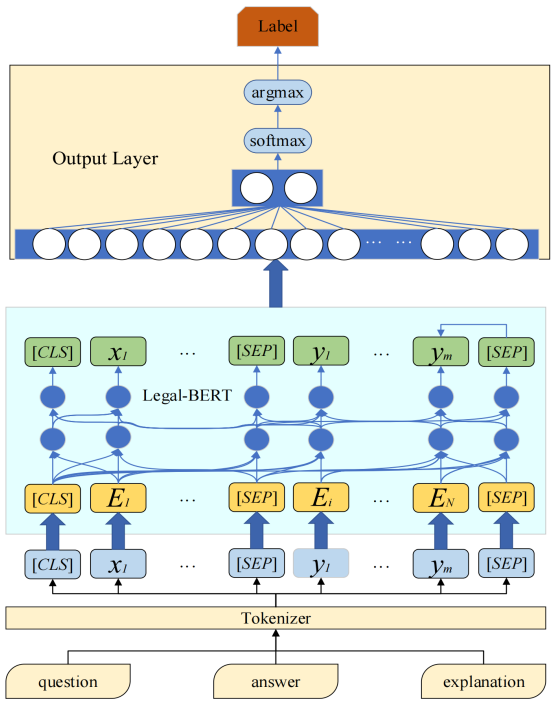


Figure 1: The structure of system

tion of the combined text, capturing the key themes and characteristics. It's a method that leverages machine learning techniques to distill the essence of a complex textual input into a single numerical value, which can be helpful in various applications such as summarization, sentiment analysis, and information retrieval.

Sliding Window Complex (SWC). In this sophisticated text processing workflow, the initial step decomposes the introductory text into discrete segments or chunks. Each chunk is meticulously constructed to include the complete question, flanked by the introduction's segments to provide context. This approach ensures that each chunk is a self-contained unit that retains the connectivity between the question and the supporting information in the introduction (Koay et al., 2021).

Subsequently, these meticulously crafted chunks are subjected to a comprehensive classification process. This process employs advanced machine learning algorithms to analyze the content of each chunk and assign it to one or more predefined categories or labels. The classification is nuanced and context-aware, considering the intricate details and subtle nuances present in the text (Kong et al., 2022).

The system employs a statistical aggregation technique to calculate the average of the predicted outputs for all the chunks. This average is a weighted sum of the individual predictions, giving more weight to chunks deemed more critical or relevant based on the specific application context.

The resulting average is a valuable metric that encapsulates the collective predictions of the model for the given question and introduction. It provides a robust summary of the model's understanding of the text, offering insights into the key themes and conclusions the model has extracted from the input. This average output can be used for various applications, such as generating summaries, making predictions, or informing decision-making processes.

2.2 Tokenizer

In many natural language processing (NLP) tasks, the original text must be processed into digital data before it can be processed by computer. Thus, the tokenizer was applied to divide the text into words and convert it into unique coding. Given a training data $\mathcal{D} = \{X^{(m)}, y^{(m)}\}_{m=1}^{M}$, $X^{(m)}$ is the processed input text. $y^{(m)}$is the corresponding ground-true label, the Bert tokenizer is applied to transform $X^{(m)}$ as,

$$X = [CLS]x_1x_2x_3...x_n[SEP]y_1y_2...y_m[SEP] \quad (1)$$

where x and y represent tokens, n and m represent the length of the first and second sentences, $[CLS]$ special mark indicates the beginning of the text sequence, $[SEP]$ indicates the separator between text sequences, respectively.

2.3 Legal-BERT Model

Legal-BERT is a specialized variant of the BERT model tailored for the legal domain, leveraging a corpus of legal text to facilitate advancements in legal natural language processing research, computational law, and legal technology applications (Chen et al., 2023). This model inherits the parameter weights from BERT-Base, ensuring a solid foundation for legal-specific tasks. In our study, we employed the pre-trained Legal-BERT model, built upon the Transformer library [1], to handle the complexities of legal language. The architecture of Legal-BERT mirrors that of the original BERT model, comprising an essential components: the Transformer encoder block (Vaswani et al., 2017). These blocks work to capture legal text's intricate

[1] https://huggingface.co/nlpaueb/legal-bert-base-uncased

patterns and nuances. The model configuration used in our experiment features 12 layers, 768 dimensions, 12 self-attention heads, and a total of 109 million parameters. This configuration balances model complexity and computational efficiency, enabling us to tackle various legal NLP challenges effectively.

Encoder block. Firstly, Legal-BERT performs the embedding operation after receiving the processed raw data. Through the above processing, we obtained token embedding, segment embedding, and position embedding (Zhang et al., 2021), followed by a series of operations to obtain $\mathbf{H}$, as follows.

$$\mathbf{H} = \mathrm{Enc}(X; \theta) \tag{2}$$

where $\mathbf{H} \in \mathbb{R}^d$ is the logits with a dimensionality of 768.

2.4 Output Layer

The BERT model has two major pretraining tasks: mask language model (MLM) and next sentence prediction (NSP), and the text implication task usually uses the NSP method to predict, that is, use the hidden layer representation of $[CLS]$ bits to predict the text classification (Ma et al., 2021). In our proposed model, the output of the model is first to use a softmax function and then perform argmax on the results after softmax to obtain $\hat{y}$,

$$\hat{y} = \mathrm{argmax}(\mathrm{softmax}(W^o\mathbf{H} + h^o)) \tag{3}$$

The training objective is to optimize the focal loss between the true and predicted labels,

$$\mathcal{L}_{FL} = \begin{cases} -(1-\hat{y}^{(m)})^{\gamma}\log(\hat{y}^{(m)}) & if\ y^{(m)} = 1 \\ -\hat{y}^{(m)^{\gamma}}\log(1-\hat{y}^{(m)}) & if\ y^{(m)} = 0 \end{cases} \tag{4}$$

where $W^o \in \mathbb{R}^d$ represents the weight of the fully connected layer, h^o represents the offset of the fully connected layer, $\mathbf{H} \in \mathbb{R}^d$ is the output representation of $[CLS]$ token in the L-th layer, γ is used to control the weight of difficult-to-classify samples, $y^{(m)}$ are respectively the true label, $\hat{y}^{(m)}$ are respectively the probability distribution of prediction.

2.5 Regularized Dropout (R-Drop)

To solve the problem of highly imbalanced data, R-Drop is added to the output layer of Legal-BERT. As shown in Figure 2, the same input can obtain two logits, $\mathbf{H}_1$ and $\mathbf{H}_2$, respectively, during the R-Drop process. Therefore, the model will output two predicted values $\hat{y}^{(1)}$ and $\hat{y}^{(2)}$, as follows.

$$\hat{y}^{(1)} = \mathrm{argmax}(\mathrm{softmax}(W^o\mathbf{H}_1 + h^o)) \tag{5}$$

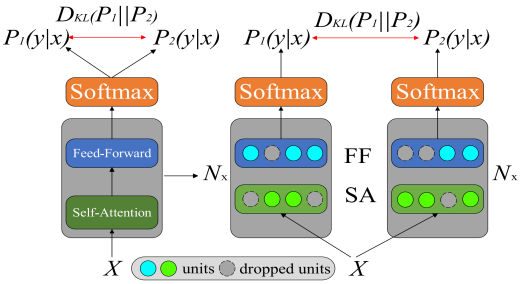


Figure 2: The structure of R-Drop

$$\hat{y}^{(2)} = \mathrm{argmax}(\mathrm{softmax}(W^o\mathbf{H}_2 + h^o)) \tag{6}$$

R-Drop uses a symmetrical Kullback-Leibler (KL) divergence to constrain $\hat{y}^{(1)}$ and $\hat{y}^{(2)}$, as follows.

$$\mathcal{L}^i{}_{KL} = \frac{1}{2}((D_{KL}(\hat{y}^{(1)}||\hat{y}^{(2)}) + D_{KL}(\hat{y}^{(2)}||\hat{y}^{(1)})) \tag{7}$$

Finally, the model will calculate the loss of two predicted values $\hat{y}^{(1)}$ and $\hat{y}^{(2)}$ using focal loss, as follows.

$$\mathcal{L}^1_{FL} = \begin{cases} -(1-\hat{y}^{(1)})^{\gamma}\log(\hat{y}^{(1)}) & if\ y^{(1)} = 1 \\ -\hat{y}^{(1)^{\gamma}}\log(1-\hat{y}^{(1)}) & if\ y^{(1)} = 0 \end{cases} \tag{8}$$

$$\mathcal{L}^1_{FL} = \begin{cases} -(1-\hat{y}^{(2)})^{\gamma}\log(\hat{y}^{(2)}) & if\ y^{(2)} = 1 \\ -\hat{y}^{(2)^{\gamma}}\log(1-\hat{y}^{(2)}) & if\ y^{(2)} = 0 \end{cases} \tag{9}$$

The training loss function for Legal-BERT is as follows.

$$\mathcal{L}_i = \mathcal{L}^1_{FL} + \mathcal{L}^2_{FL} + \mathcal{L}^i_{KL} \tag{10}$$

3 Experimental Result

Datasets. The Legal Argument Reasoning Task in Civil Procedure shared task data set is composed of three CSV files: the size of the training set train.csv sorted by expert comments is 666, the size of the developing set dev.csv is 84, the size of test set test.csv is 98. The data part of the train and dev set mainly includes idx, question, answer, label, analysis, complete analysis, and explanation. The data part of the test set mainly includes idx, question, answer, and explanation. Idx is used to represent the number of each sample. The question is made in the context of the content of the explanation. The answer is a candidate answer in the sample. Label indicates whether the question and candidate

Question: 8. Technical fouls. Eban brings suit against Lorenzo for interference with business relations. Eban's lawyer, Darrow, calls Lorenzo's lawyer, Sadecki, and tells her that he is filing suit that afternoon. In which of the following scenarios may the case proceed without formal service of process?
Answer: Darrow files the complaint and mails a copy of it by certified mail to Lorenzo with two copies of a proper request for waiver. He receives the green postal receipt back in the mail. Darrow files the postal receipt with the court.
BERT Predicted label: 1
Legal-BERT Predicted label: 0
True label: 0

Question: 8. Technical fouls. Eban brings suit against Lorenzo for interference with business relations. Eban's lawyer, Darrow, calls Lorenzo's lawyer, Sadecki, and tells her that he is filing suit that afternoon. In which of the following scenarios may the case proceed without formal service of process?
Answer: Darrow files the complaint and mails a copy of it by certified mail to Lorenzo with two copies of a proper request for waiver. He does not send a summons, signed and sealed by the court, with the waiver request. He receives the signed waiver form back and files it with the court.
BERT Predicted label: 0
Legal-BERT Predicted label: 1
True label: 1

Figure 3: Examples of different models on the dev set

answer match, 0 for mismatch, and 1 for matching. Analysis and complete analysis are used for experimenters to understand why the label is 0 or 1. Explanation is used to indicate the subject of the sample to which it belongs.

Evaluation Metrics. The Legal Argument Reasoning Task in Civil Procedure shared tasks are evaluated using the standard evaluation indicators, including Macro F_1-*score* and Accuracy. The submissions of all teams are ranked according to the F_1-*score*. The metrics will be calculated as follows.

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (11)$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (12)$$

$$F_1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (13)$$

Implementation Details. Initially, explanation and question are concatenated when processing data. The BERT (Devlin et al., 2018) is used as the first model to solve this task. However, without any treatment, the predicted value of the BERT is all 0, and the effect is not ideal. Next, we used the larger models RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020), but the predictions and $F1$_scores were identical to BERT. Due to the

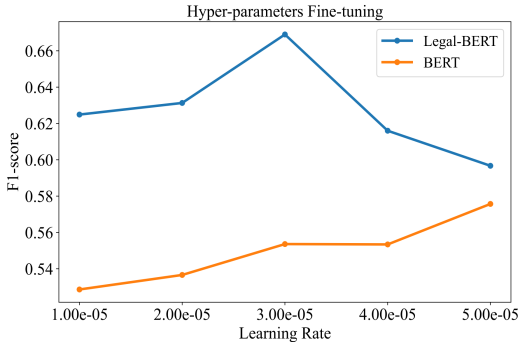


Figure 4: The performance of different learning rates on the F_1-*score*

extreme data imbalance, we found that the cross-entropy loss function could not calculate the loss correctly. Therefore, we changed the loss function for BERT, RoBERTa, and DeBERTa to focal loss and dice loss. The results show that modifying the loss function can slightly improve the score, but the effect is not ideal. To solve the problem of extreme data imbalance further, we change their loss functions to focal loss and dice loss (Li et al., 2020) based on supervised contrastive learning (SCL) (Khosla et al., 2020) and R-Drop. The results show that the combination of pairs can effectively solve the problem of extreme data imbalance, and the score has also been significantly improved. During the experiment, we found that due to the large number of proprietary legal terms in the data text, the above model could not fully segment professional vocabulary using the corresponding tokenizer. Therefore, we believe that the Legal-BERT is the most suitable choice. As expected, Legal-BERT has achieved good results in adding R-Drop and focal Loss technologies, as shown in Figure 3.

Hyper-parameters Fine-tuning. We adjusted different learning rates and epochs to adapt to different models to achieve the expected results. Legal-BERT is better than BERT regardless of the learning rate, as shown in Figure 4. The optimal F_1-*score* was found at 4 with the batch size constantly changing, as shown in Figure 5. We set the best parameters in the final submitted results: warmup steps are 10, weight decay is 0.01, the learning rate is 3e-5, train batch size is 4, and epoch is 100.

Comparative Results and Discussion. The test is first carried out on the development set, whose size is 84. Facing the different predicted results of other models and Legal-BERT, it is clear that Legal-BERT performs better. Regardless of the

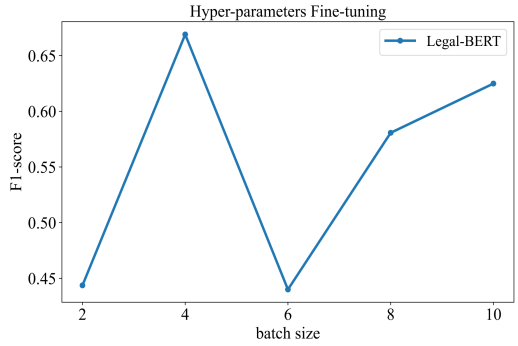


Figure 5: The performance of different batch sizes on the F_1-*score*

Model	Loss	F_1-*score*	Accuracy
BERT	Cross-Entropy	0.4437	0.7976
RoBERTa	Cross-Entropy	0.4437	0.7976
DeBERTa	Cross-Entropy	0.4437	0.7976
Legal-BERT	Cross-Entropy	0.4437	0.7976
BERT	Focal Loss	0.4688	0.8095
RoBERTa	Focal Loss	0.4437	0.7976
DeBERTa	Focal Loss	0.4956	0.7976
Legal-BERT	Focal Loss	0.5599	0.6548
BERT	Dice Loss	0.5468	0.6548
RoBERTa	Dice Loss	0.4830	0.7738
DeBERTa	Dice Loss	0.4830	0.7738
Legal-BERT	Dice Loss	0.4943	0.7421

Table 1: models and methods.

model, as long as the loss function is cross entropy, the final predicted value will be 0. Both dice loss and focal loss can solve the problem of imbalance in data, but focal loss is more effective. When SCL and R-Drop were introduced, R-Drop achieved significantly better results. Legal-BERT can deal with legal vocabulary more thoroughly than other models. Overall, Legal-BERT+R-Drop+focal Loss is the best combination obtained after experiments. The F_1-*score* obtained from the experiments of several models and methods is summarized in Table 1, Table 2, and Table 3, and the result of the best submission is shown in Table 4. Although the sliding window approach helps alleviate the token limitations of Legal-BERT, models specifically designed to handle longer documents, such as Longformer (Beltagy et al., 2020) or Big Bird (Zaheer et al., 2020), might offer superior efficiency. In the future, our team will also use the above model to solve the problem of long text.

4 Conclusion

In this research paper, we introduce a system submitted for evaluation in SemEval-2024 Task 5. Leveraging the powerful pre-trained Legal-BERT

Model	Loss	F_1-*score*	Accuracy
BERT + SCL	Cross-Entropy	0.4437	0.7976
RoBERTa + SCL	Cross-Entropy	0.4437	0.7976
DeBERTa + SCL	Cross-Entropy	0.4437	0.7976
Legal-BERT + SCL	Cross-Entropy	0.4437	0.7976
BERT + SCL	Focal Loss	0.5625	0.6428
RoBERTa + SCL	Focal Loss	0.5460	0.8095
DeBERTa + SCL	Focal Loss	0.4247	0.7381
Legal-BERT + SCL	Focal Loss	0.5296	0.6706
BERT + SCL	Dice Loss	0.4892	0.7302
RoBERTa + SCL	Dice Loss	0.4437	0.7976
DeBERTa + SCL	Dice Loss	0.4437	0.7976
Legal-BERT + SCL	Dice Loss	0.5299	0.6508

Table 2: models and methods.

Model	Loss	F_1-*score*	Accuracy
BERT + R-Drop	Cross-Entropy	0.4437	0.7976
RoBERTa + R-Drop	Cross-Entropy	0.4437	0.7976
DeBERTa + R-Drop	Cross-Entropy	0.4437	0.7976
Legal-BERT + R-Drop	Cross-Entropy	0.4437	0.7976
BERT + R-Drop	Focal Loss	0.5637	0.6746
RoBERTa + R-Drop	Focal Loss	0.4437	0.7976
DeBERTa + R-Drop	Focal Loss	0.5650	0.6510
Legal-BERT + R-Drop	Focal Loss	0.6690	0.8210
BERT + R-Drop	Dice Loss	0.4824	0.6310
RoBERTa + R-Drop	Dice Loss	0.4437	0.7976
DeBERTa + R-Drop	Dice Loss	0.5155	0.6310
Legal-BERT + R-Drop	Dice Loss	0.4437	0.7976

Table 3: models and methods.

F_1-*score*	Accuracy
0.6166	0.6837

Table 4: best submission result.

model as its foundation, our system underwent essential modifications to enhance performance. Specifically, we refined the loss function and incorporated the R-Drop technique to determine the alignment between questions and their corresponding answers accurately. The empirical results obtained from our experiments demonstrate the effectiveness of our proposed system, showcasing its strong performance capabilities. However, when benchmarked against the leading systems in the competition, it becomes evident that there are still notable areas for further improvement. Looking ahead, we are eager to explore the integration of alternative legal-specific models and innovative length text processing strategies. By pursuing these avenues, we aim to achieve even more promising results that can contribute significantly to advancing the field.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Natural Sci-

ence Foundation of China (NSFC) under Grant Nos.61966038 and 62266051.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Yu Chen, You Zhang, Jin Wang, and Xuejie Zhang. 2023. YNU-HPCC at SemEval-2023 task 6: LEGAL-BERT based hierarchical BiLSTM with CRF for rhetorical roles prediction. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2075–2081, Toronto, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Jia Jin Koay, Alexander Roustai, Xiaojin Dai, and Fei Liu. 2021. A sliding-window approach to automatic creation of meeting minutes. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 68–75, Online. Association for Computational Linguistics.

Jun Kong, Jin Wang, and Xuejie Zhang. 2022. Hierarchical bert with an adaptive fine-tuning strategy for document classification. *Knowledge-Based Systems*, 238:107872.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Xinge Ma, Jin Wang, and Xuejie Zhang. 2021. YNU-HPCC at SemEval-2021 task 11: Using a BERT model to extract contributions from NLP scholarly articles. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 478–484, Online. Association for Computational Linguistics.

ANDRÉ VALENTE, JOOST BREUKER, and BOB BROUWER. 1999. Legal modeling and automated reasoning with on-line. *International Journal of Human-Computer Studies*, 51(6):1079–1125.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

You Zhang, Jin Wang, and Xuejie Zhang. 2021. Personalized sentiment classification of customer reviews via an interactive attributes attention model. *Knowledge-Based Systems*, 226:107135.

TECHSSN at SemEval-2024 Task 10: LSTM-based Approach for Emotion Detection in Multilingual Code-Mixed Conversations

Ravindran V, Shreejith Babu G, Aashika Jetti
Rajalakshmi Sivanaiah, Angel Deborah S, Mirnalinee T T, Milton R S
Department of Computer Science and Engineering
Sri Sivasubramaniya Nadar College of Engineering
Chennai - 603110, Tamil Nadu, India
{ravindran2213003, shreejithbabu2213006, aashika2210193}@ssn.edu.in,
{rajalakshmis, angeldeborahs, mirnalineett, miltonrs}@ssn.edu.in

Abstract

Emotion Recognition in Conversation (ERC) in the context of code-mixed Hindi-English interactions is a subtask addressed in SemEval-2024 as Task 10. We made our maiden attempt to solve the problem using natural language processing, machine learning and deep learning techniques, that perform well in properly assigning emotions to individual utterances from a predefined collection. The use of well-proven classifier such as Long Short Term Memory networks improve the model's efficacy than the BERT and Glove based models. However, difficulties develop in the subtle arena of emotion-flip reasoning in multi-party discussions, emphasizing the importance of specialized methodologies. Our findings shed light on the intricacies of emotion dynamics in code-mixed languages, pointing to potential areas for further research and refinement in multilingual understanding.

1 Introduction

The ultimate objective of this task EDiReF is to make progress in the field of conversational emotion recognition and reasoning. Analyzing emotions in natural language offers measurable understandings within the typically subjective domain of expressive language, connecting disciplines like psychology, cognition, and linguistics. This focuses on code-mixed Hindi-English dialogues, creating a unique and challenging linguistic setting in which participants must decipher the complexities of emotion recognition (ERC) and emotion flip reasoning (EFR). Code-mixing, or the intentional use of different languages within a single conversation, complicates the process and necessitates novel techniques for effective emotion recognition. This kind of collaborative work is critical because it reflects the changing environment of communication, where diversified language use needs complex models capable of understanding emotions in multilingual discussions resulting in overall development.

Our approach which focuses on the Emotion Recognition in Conversation (ERC) subtask, employs a sophisticated strategy based on deep learning methodologies suited to the intricacies of code-mixed Hindi-English (HI-EN) talks. At its core, our solution employs a Bidirectional Long Short-Term Memory (Bi-LSTM) network, an architecture capable of capturing intricate sequential connections inside text. We chose LSTM as LSTM networks are designed to overcome the limitations of traditional recurrent neural networks (RNNs) by mitigating the vanishing gradient problem and LSTM's ability to retain and selectively update information over time can lead to more accurate predictions of emotional states. We prioritized preprocessing, which includes tokenization and sequence padding, to help the model understand speech contexts. To enhance linguistic representation, the embedding layer is initialized with pre-trained word embeddings. For regularization and addressing overfitting issues, strategically placed dropout layers are incorporated. The training process utilizes categorical cross-entropy loss and the Adam optimizer.

The model showcased proficiency in precisely attributing emotions to individual utterances, showcasing its capability to decipher intricate emotional expressions in a multilingual context. From a quantitative standpoint, our system achieved an accuracy of 0.378 in sentiment analysis and secured the 23rd position among the competing teams. These findings not only provide insights into the model's strengths and areas for improvement but also highlight the importance of specialized mechanisms for complicated emotional connections in multilingual communication such as expanding datasets to encompass a broader range of linguistic and cultural contexts, as well as areas for future research

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 763–769

2 Background

The task at hand focuses on comprehending and categorizing emotions presented in code-mixed Hindi-English interactions. In this context, the input comprises dialogues where individuals communicate in both Hindi and English. The primary aim is to analyze each segment of these interactions and assign a precise emotion from a predefined set to capture nuanced sentiments. For example, throughout a conversation, people may display a variety of emotions at different points, such as joy, sorrow, or rage.

The task comprises two datasets centered around Multi-modal Sarcasm Detection and Humor Classification (MaSac) for sub-tasks 1 and 2, and Multimodal Emotion Lines (MELD) for subtask 3, but this paper majorly focuses on sub-task 1. The datasets for this task consist of code-mixed conversations, reflecting the real-world scenario where individuals seamlessly blend Hindi and English while communicating. The genre of the conversations may vary, encompassing diverse topics and contexts to ensure a comprehensive understanding of emotion dynamics in code-mixed language interactions. The size of the datasets is not explicitly mentioned, but it likely involves a substantial amount of annotated dialogues to train and evaluate emotion recognition models effectively. The dataset consisted of four parameters the episode name, the speakers list, the utterances, and the emotions mapped to the respective utterances. The emotions mapped to the utterances provided a standard for assessing models' performance in recognizing emotional expressions. For example, the utterance "ok, chalo rosesh chalo bahar" was mapped to the emotion "Contempt".

3 Related Work

Arora et al. (Arora et al., 2016) explores the capabilities of deep neural networks (DNN) using rectified linear units (ReLU). It introduces an algorithm for training a ReLU DNN with one hidden layer to global optimality with polynomial runtime in data size. The paper also improves lower bounds for approximating ReLU deep net functions and provides gap theorems for smoothly parametrized families of "hard" functions. Notably, it demonstrates the existence of functions requiring k^3 total nodes in a ReLU DNN with k^2 hidden layers, shedding light on the network's complexity.

Anshul Wadhawan and Akshita Aggarwal et al. (Wadhawan and Aggarwal, 2021) presents a Transformer-based approach for detecting emotions in code-mixed tweets. They introduce a Hinglish dataset, use bilingual word embeddings, and experiment with various models, including CNNs, LSTMs, and transformers like BERT. The BERT model achieves the best accuracy at 71.43%. The paper highlights the importance of emotion detection in social media and multilingual contexts, providing a valuable annotated dataset for future research.

Shivani Kumar et al. (Kumar et al., 2023a; Bedi et al., 2021) delved into Emotion Flip Reasoning (EFR) in multiparty conversations, showcasing state-of-the-art performance against baselines. Their research highlights the significance of EFR in enhancing empathetic response generation and understanding emotional dynamics in conversational settings, thus addressing the gap and providing insights into how specific remarks or expressions affect listeners.

Deepanshu Vijay et al. (Vijay et al., 2018) addresses emotion prediction in Hindi-English code-mixed social media text. They introduce a corpus from Twitter annotated with emotions and source languages. The paper proposes a supervised classification system using machine learning techniques and diverse features for emotion detection, contributing to resources for Hindi-English code-mixed text analysis in multilingual contexts.

In a parallel domain, a study focused on emotion analysis in low resource language Tamil is done by Varsini et al. (S et al., 2022). They have employed a lexicon-based approach and transformer models, utilizing dictionaries of words labeled with emotions. This research specifically addresses the challenges of extracting emotions from low resource texts in social media contexts, offering valuable insights.

Contextual emotion detection is executed using gaussian model and ensemble model by Angel Deborah et al. (Deborah et al., 2022; Angel Deborah et al., 2020). The challenge of contextual emotion detection in natural language processing has been addressed, emphasizing the difficulty for both machines and humans to accurately detect emotions like sadness or disgust in a sentence without sufficient context. The study underscores the growing importance of providing sensible responses in text messaging applications, where digital agents play a prominent role. The research showcases the efficacy of a Gaussian process for detecting contextual

emotions within sentences, comparing its performance with Decision Tree and ensemble models, including Random Forest, AdaBoost, and Gradient Boost.

Emotion recognition in Hindi-English code-mixed data, as explored in relevant papers, employs models like BERT, RoBERTa, CNNs, and LSTMs. The challenges highlighted align with our work, emphasizing the importance of addressing code-mixing complexities and the scarcity of annotated datasets. Similarly, in corpus creation for emotion prediction, the focus on a Twitter-based annotated corpus resonates with our efforts. The shared emphasis on overcoming linguistic diversity and cultural nuances underscores the mutual pursuit of enhanced accuracy in emotion recognition, urging continued research in these aspects.

4 System Overview

To optimise efficiency, we methodically integrated numerous critical algorithms and modelling decisions into our sentiment analysis model.

4.1 Data Preprocessing

4.1.1 Text Cleaning and Tokenization

The initial phase of our sentiment analysis model required thorough dataset preprocessing. The dialogue data (Kumar et al., 2023b, 2024) includes annotations for various emotions expressed by the speakers. These utterances are in both Hindi and English. We used Python's regular expressions and popular natural language processing packages to apply text cleaning techniques. Special characters, numerals, and unnecessary spaces were deleted. The cleaned text was then tokenized with TensorFlow Keras and (Arora, 2020) Indic NLP packages as It provides language-specific tokenization and other preprocessing functionalities tailored to languages spoken in the Indian subcontinent, improving the model's understanding of linguistic nuances.

4.1.2 Language-specific Tokenization

The dataset's multilingual composition necessitated the use of a complex tokenization technique. English utterances were tokenized with (Loper and Bird, 2002) NLTK's word tokenizer which breaks the text into individual words while preserving English language semantics, whereas Hindi text was tokenized with the Indic NLP package. It generates tokens by separating the text into its constituent words or tokens based on the identified boundaries. The goal of this multilingual tokenization technique was to identify and record language-specific patterns that were present across the dataset. The model is able to absorb and comprehend the unique linguistic aspects of both languages in the dataset because of this bilingual tokenization technique.

4.1.3 Stop Word Removal

In order to enhance the model's attention towards meaningful content, we systematically removed stopwords from both Hindi and English text. This important preprocessing step allowed for a more detailed understanding of the underlying sentiment by removing unnecessary noise and refining the raw data. In addition, we consistently converted all text to lowercase for uniformity and better generalization. This approach to preprocessing contributes to the model's robust performance in capturing the intricacies of emotion in code-mixed interactions.

4.1.4 Data Splitting:

The preprocessed data was divided into two parts: the training and the testing sets using the `train_test_split` function from the `scikit-learn` library. This ensures that the model's performance can be evaluated on unseen data, facilitating a thorough assessment of its generalization ability. In this approach, we could closely examine the extent to which our model could process novel, unseen data and see whether it could apply the knowledge it gained to a wider context.

4.2 Model Architecture

4.2.1 Embedding Layer

At the center of our sentiment analysis model is the Embedding layer. It takes tokenized words and transforms them into smart vectors. This layer, configured with an input dimension of `max_words`, an output dimension of 128, and an input length of `max_len`, converts tokenized input sequences into dense vectors that capture semantic associations between words. This layer essentially helps the model understand the deep connections and meanings between words in the input sequences.

4.2.2 Bidirectional LSTM

To capture the sequential dependencies in language effectively, we incorporated a Bidirectional Long Short-Term Memory (LSTM) layer (Staudemeyer and Morris, 2019). With 64 units, this bidirectional architecture facilitates the model in understanding

contextual relationships in both forward and backward directions.

4.2.3 Dense and Dropout Layers

A dense layer of 64 units was inserted sequentially, followed by Rectified Linear Unit (ReLU) activation (Arora et al., 2016). This layer, along with a dropout layer with a rate of 0.5, dramatically improved the model's ability to recognize complicated patterns while reducing overfitting. This is also adds a moderation in the learning process.

4.2.4 Output Layer

Using the softmax activation function (Sharma et al., 2017), the model's output layer successfully classified emotions into distinct categories as it converts the raw output scores of the model into probabilities, indicating the correct classification for the input. This categorical method enabled a more sophisticated comprehension of the diverse attitudes exhibited in the dataset.

In the final act, our model showcased its classification prowess through the output layer. With a touch of softmax activation function (Sharma et al., 2017), it skillfully categorized emotions into distinct categories. This categorical wizardry allowed our model to attain a nuanced understanding of the diverse attitudes presented in the dataset.

4.3 Model Training

4.3.1 Loss Function and Optimizer

The model was built using categorical cross entropy loss function and the Adam optimizer (Zhang and Sabuncu, 2018), known for its efficiency in handling sparse gradients. The rationale behind selecting the categorical crossentropy loss function and the Adam optimizer lies in their proven track record of effectiveness in sentiment analysis endeavors. Categorical cross-entropy performs well in circumstances with several classes, precisely meeting the requirements of sentiment classification with distinct emotion labels. By making the model allocate higher probabilities to the correct class, this loss function fosters more accurate sentiment predictions. This combination was intended to successfully optimize the model's weights, resulting in a robust learning process during training.

4.3.2 Training Parameters

A batch size of 32 and five epochs were used in the training procedure. This configuration produced the ideal training length by striking a balance between model convergence and processing efficiency.

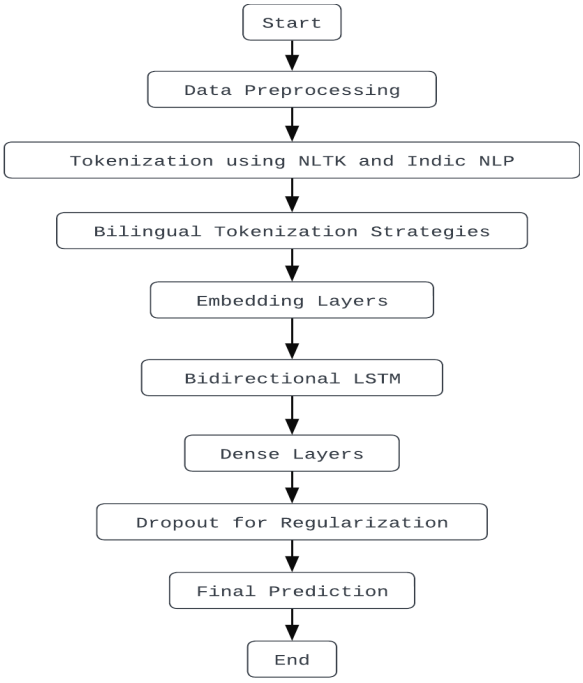


Figure 1: Model Process

4.4 Model Evaluation

4.4.1 Perfomance Metrics

The model's performance was evaluated using metrics including accuracy, loss, and weighted F1 score after training. These metrics provided a comprehensive understanding of how well the model identified emotions in textual input.

4.4.2 Prediction on the Test Set

Emotions on a test set were predicted using the trained model. This step consisted of applying the model to the preprocessed test data and decoding the predicted labels for further examination. The overall process of working is shown in Figure 1.

5 Experimental Setup

The data split used in the given code divides the dataset into training and testing sets. The split ratio is 80% for training data and 20% for testing data, as stated by the code's `test_size` argument of 0.2 to prevent overfitting and perform better on new data.

The learning rate is a crucial hyperparameter that determines the step size during the optimization

process.The learning rate, a key factor in the optimization process, was explored with values such as 0.01, 0.001, and 0.0001. To mitigate overfitting, dropout rates were varied, including options like 0.2 and 0.5

The number of LSTM units in the Bidirectional LSTM layer, a crucial aspect of model capacity, was adjusted with values like 32, 64, and 128. Additionally, the impact of different batch sizes (16, 32, 64) on convergence and computational efficiency was systematically explored.

In our experimental setup, we harnessed the power of scikit-learn for seamless implementation of various machine learning algorithms, handling data preprocessing tasks, and evaluating performance using diverse metrics. The NLTK library, an essential component, efficiently managed critical natural language processing functions, including tokenization, stopwords removal, and stemming.

To ensure model persistence and flexibility, we adopted joblib, a tool adept at saving and loading trained models. Moreover, our approach integrated external tools like NLTK Indic NLP, Scikit-Learn, and TensorFlow to elevate specific components of our sentiment analysis model. This encompassed optimizing tokenization, refining data splitting techniques, streamlining preprocessing steps, and conducting rigorous model evaluations, all contributing to the robustness and effectiveness of our experimental framework.

6 Results

The task is evaluated using the following performance metrics: precision, recall, accuracy and F1-score.

Recall indicates the classifier's ability to identify positive instances accurately and accuracy is defined as the ratio of the correctly predicted instances to the total number of instances in a dataset. It acts as a straightforward for the model's correctness while precision is a measure of how accurate the positive predictions are.

$$Precision = \frac{\text{TP}}{TP + \text{FP}} \quad (1)$$

$$Recall = \frac{\text{TP}}{TP + \text{FN}} \quad (2)$$

$$Accuracy = \frac{\text{TP + TN}}{TP + \text{TN} + \text{FP} + \text{FN}} \quad (3)$$

Approach	Accuracy
Word GloVe	0.35
Dist-Bert	0.30
LSTM Model	0.378

Table 1: Comparison of Accuracy for Different Approaches

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

$$Weighted\ F1\ Score = \sum_{i=1}^{N} W_i \cdot F1\ Score_i \quad (5)$$

The sentiment analysis model demonstrated the performance on the training dataset, achieving an accuracy of 0.39 and weighted F1 score of 0.38. The classification report is shown in Figure 2. On the test set, the model maintained the performance with an accuracy of approximately 0.378 and a weighted F1 score of 0.34 In the competition results, the model secured 23rd position. With LSTM approach, our model achieved an accuracy of 0.378 and secured the 23th position on the rankings. The comparison of various developed models are listed in Table 1.

We used a variety of approaches to improve the accuracy of sentiment analysis. One important approach was using pre-trained word embeddings, such as Word Glove (Rezaeinia et al., 2019), which captures semantic associations between words. The use of Word GloVe embeddings enabled a decent comprehension of contextual nuances, which contributed to increased sentiment analysis results with a F1 Score of about 0.35. While we initially explored the use of Word GloVe embeddings, we found that other methodologies yielded better results for our specific task. Therefore, we transitioned away from Word GloVe embeddings and pursued alternative approaches that demonstrated improved performance in managing the challenges associated with dual tokenization in mixed-language conversations. We also attempted to develop DistBERT, a transformer model, but faced a difficult case in which its integration resulted in an underwhelming F1 score of 0.30. This unexpected outcome spurred a rethinking of the implementation, prompting us to investigate alternate tactics and optimisations to improve the model's effectiveness.

	precision	recall	f1-score	support
anger	0.21	0.21	0.21	168
contempt	0.15	0.14	0.15	119
disgust	0.00	0.00	0.00	28
fear	0.14	0.11	0.12	83
joy	0.35	0.36	0.35	313
neutral	0.53	0.58	0.55	801
sadness	0.12	0.11	0.11	107
surprise	0.31	0.23	0.26	83
accuracy			0.39	1702
macro avg	0.23	0.22	0.22	1702
weighted avg	0.38	0.39	0.38	1702

Figure 2: Classification Report

Owing to the model's training on the restricted quantity of available data, it exhibits a bias towards specific emotions. "Okay chaliye dad, mein aapko bahar fenk kar aata hun!" is an example of a statement that should be predicted as "Joy," instead it is predicted as "neutral." The reason for the model's behaviour is that a lot of utterances are mapped to the neutral emotion; as a result, when a model is trained on this kind of data, it naturally becomes biased towards such emotion types.

In addition, a thorough analysis of the classification reports shed additional light on the theory on how class imbalances affect the model's functionality. As can be seen from the performance measures that were previously addressed, there are significant differences in the weighted and macro F1-scores, even if the classifiers' accuracy is the same for all datasets that were used. In particular, the sentiment analysis model performs noticeably better on the Hindi-English code-mixed dataset than on the Hindi and English monolingual datasets, highlighting the difficulties caused by the imbalances in the latter. This limitation is particularly notable for emotions with limited representation in the training data, emphasizing the need for strategies to address imbalances and improve the models' robustness across diverse linguistic scenarios.

7 Conclusion

We lay out a sentiment analysis system that can handle Hindi-English talks with mixed codes. Recognising and rationalising emotions in a bilingual environment was the main goal. The results show that the participants performed competitively in terms of emotion perception and reasoning, especially when there was frequent language change. The model using LSTM layers, NLTK Indic NLP provided the best result of 0.36 F1 score. Subtle emotional cues and particular code-mixing patterns continue to provide difficulties, nevertheless. Our method is noteworthy for its hybrid approach, which makes use of sentiment analysis, contextual embedding techniques, and language models that have already been trained and refined using code-mixed datasets.

It is still imperative to address specifics in low-resource languages in future development. Techniques like compiling lists of language-specific stop words have shown to be effective. Moreover, the effect of class disparities on the model's functionality is recognized. Furthermore, the effect of class imbalances on the performance of the model is recognized. Subsequent research could investigate customized approaches, including data enrichment or clustering techniques, to address these imbalances and improve the model's flexibility in a variety of language circumstances unique to our code.

References

S Angel Deborah, S Rajalakshmi, S Milton Rajendram, and TT Mirnalinee. 2020. Contextual emotion detection in text using ensemble learning. In *Emerging Trends in Computing and Expert Technology*, pages 1179–1186. Springer.

Gaurav Arora. 2020. inltk: Natural language toolkit for indic languages. *arXiv preprint arXiv:2009.12534*.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. 2016. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*.

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2021. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Transactions on Affective Computing*, 14(2):1363–1375.

S Angel Deborah, Rajendram S Milton, TT Mirnalinee, and S Rajalakshmi. 2022. Contextual emotion detection on text using gaussian process and tree based classifiers. *Intelligent Data Analysis*, 26(1):119–132.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023a. Emotion flip reasoning

in multiparty conversations. *IEEE Transactions on Artificial Intelligence*.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023b. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Seyed Mahdi Rezaeinia, Rouhollah Rahmani, Ali Ghodsi, and Hadi Veisi. 2019. Sentiment analysis based on improved pre-trained word embeddings. *Expert Systems with Applications*, 117:139–147.

Varsini S, Kirthanna Rajan, Angel S, Rajalakshmi Sivanaiah, Sakaya Milton Rajendram, and Mirnalinee T T. 2022. Varsini_and_Kirthanna@DravidianLangTech-ACL2022-emotional analysis in Tamil. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 165–169, Dublin, Ireland. Association for Computational Linguistics.

Sagar Sharma, Simone Sharma, and Anidhya Athaiya. 2017. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316.

Ralf C Staudemeyer and Eric Rothstein Morris. 2019. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Corpus creation and emotion prediction for hindi-english code-mixed social media text. In *Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: student research workshop*, pages 128–135.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in hindi-english code-mixed data: A transformer based approach. *arXiv preprint arXiv:2102.09943*.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.

UIR-ISC at SemEval-2024 Task 3: Textual Emotion-Cause Pair Extraction in Conversations

Hongyu Guo, Xueyao Zhang, Yiyang Chen, Lin Deng*, Binyang Li
Lab of Intelligent Social Computing
University of International Relations, Beijing, China
{chloe_guo,Zhang_X_Y,uiryangyc0114,denglin,byli}@uir.edu.cn

Abstract

The goal of Emotion Cause Pair Extraction (ECPE) is to explore the causes of emotion changes and what causes a certain emotion. This paper proposes a three-step learning approach for the task of Textual Emotion-Cause Pair Extraction in Conversations in SemEval-2024 Task 3, named ECSP. We firstly perform data preprocessing operations on the original dataset to construct negative samples. Secondly, we use a pre-trained model to construct token sequence representations with contextual information to obtain emotion prediction. Thirdly, we regard the textual emotion-cause pair extraction task as a machine reading comprehension task, and fine-tune two pre-trained models, RoBERTa and SpanBERT. Our results have achieved good results in the official rankings, ranking 3rd under the strict match with the *Strict F1-score* of 15.18%, which further shows that our system has a robust performance.

1 Introduction

Emotions are innate to humans and significantly affect people's social interactions, decision-making, and cognition. People are becoming more interested in developing human-like reactions as social media evolves. Therefore, the recognition of emotions in the text is an important topic in natural language processing and its applications (Zhao et al., 2016). In addition to emotion recognition, the research on the cause behind emotions in conversation scenarios is more complex, such as customer support, mental health care, human-computer interaction, etc (Wang et al., 2023b). Thus, it is important to recognize the potential cause behind an individual's emotional state, i.e., Emotion Cause Analysis (ECA). [1]

In recent research, Xia and Ding (2019) proposed the Emotion Cause Pair Extraction (ECPE) task, which is used to automatically predict emotions in documents and recognize the corresponding causes of those emotions. This task has attracted attention from a number of academics (Ding et al., 2020; Wei et al., 2020; Chen et al., 2020). However, the ECPE task studies the emotion-cause relationship of specific events in the document, while in the conversational scene, due to the interaction of multiple speakers, the dialogue contains more diverse and richer emotional expressions, which makes the conversation continue to advance as the conversation progresses. Emotions are also constantly changing, and the emotion of one utterance may be caused by multiple utterances.

In this paper, we propose a three-step learning approach, **E**motion-**C**ause-**S**pan **P**air Extraction in Conversation (**ECSP**), for Subtask 1 of SemEval-2024 Task 3: Textual Emotion-Cause Pair Extraction in Conversations. ECSP consists of three modules: the data preprocessing module, the emotion classification module, and the textual emotion-cause pair extraction module. We first preprocessed the dataset to obtain a large number of negative examples. Then, the pre-trained model BERT is used to construct token sequence representations with contextual information that are fed into a feed-forward neural network layer for emotion prediction. In the textual emotion-cause pair extraction module, in order to obtain causal span, we fine-tuned pre-trained models such as RoBERTa and SpanBERT to make it a machine reading comprehension (MRC) task (Poria et al., 2021).

In the official ranking, our team ranked 3rd under the strict match with the *Strict F1-score* of 15.18%, and ranked 7th under the Proportional match with the *Proportional F1-score* of 19.63%.

*Corresponding author

[1] Description of the task by the organizer of SemEval-2024 Task 3

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 770–776

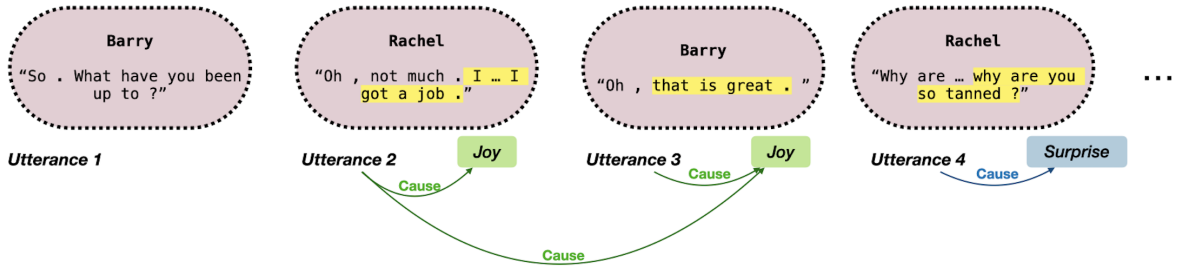


Figure 1: Description of the task of textual emotion-cause pair extraction in conversations.

2 Background

2.1 Task Definition

As shown in Figure 1, the task of textual emotion-cause pair extraction in conversations aims to extract all emotion-cause pairs in a given conversation based entirely on text and mark the specific causal span of the emotion cause (Wang et al., 2024).

Input: A conversation containing the speaker and the text of each utterance. Represented as the content in the pink rectangular box in Figure 1.

Output: All predicted emotion-cause pairs, where each pair contains an emotion utterance along with its emotion category and the textual cause span in a specific cause utterance. The utterance pointed by the curve to the emotion in the Figure 1 is the cause utterance of the emotion, and the yellow background text fragment is a specific textual cause span.

2.2 Related Work

Emotions always play a vital role in information exchange, from the communication between human individuals in the real world to the human-computer interaction in the virtual world. Recognizing emotion categories in text is an essential task in NLP and its applications (Zhao et al., 2016). In addition, the causes of emotions play a key role in human-computer interaction and customer service systems, which can provide important information on the reason for any emotion changes.

The aim of Emotion Cause Extraction (ECE) is to explore the causes of emotion changes and what causes a certain emotion (Chen et al., 2010). Xia and Ding (2019) reformed ECE into ECPE (Emotion-Cause Pair Extraction), aiming to extract potential emotions and corresponding causes from documents simultaneously.

Since ECPE does not fully consider the correlation between emotional utterances and causal utterances and the limited availability of background, Shan and Zhu (2020) proposed an Inter-EC model with self-attention, which optimized the interactive multi-task network model. Cheng et al. (2021) reconstructed the emotion-cause pair extraction task into the classification problem of candidate sentence pairs and proposed a goal-oriented, unified sequence-to-sequence model. Poria et al. (2021) constructed a dialogue-level dataset RECCON and introduced a task highly relevant for (explainable) emotion-aware to address causal span extraction and causal emotion entailment.

3 System Overview

In order to implement the task of textual emotion-cause pair extraction in conversations, we have designed the ECSP approach, which contains three main modules, namely data preprocessing, emotion classification, and textual emotion-cause pair extraction.

Firstly, in the data preprocessing module, the dataset is preprocessed to obtain a large number of negative samples. Then the pre-trained model BERT is used to convert token sequences with contextual information in the conversation into semantic representations and predict emotions in the emotion classification module. Finally, textual emotion-cause pairs are extracted based on the predicted emotions in the textual emotion-cause pair extraction module.

The overall architecture of ECSP system is shown in Figure 2, and the detailed description for each part is presented as follows.

3.1 Data Preprocessing Module

Since the original dataset only contains positive examples, i.e., utterances containing emotions, which are annotated using causal spans extracted from the historical context of the conversation, we designed the data preprocessing module to provide a large

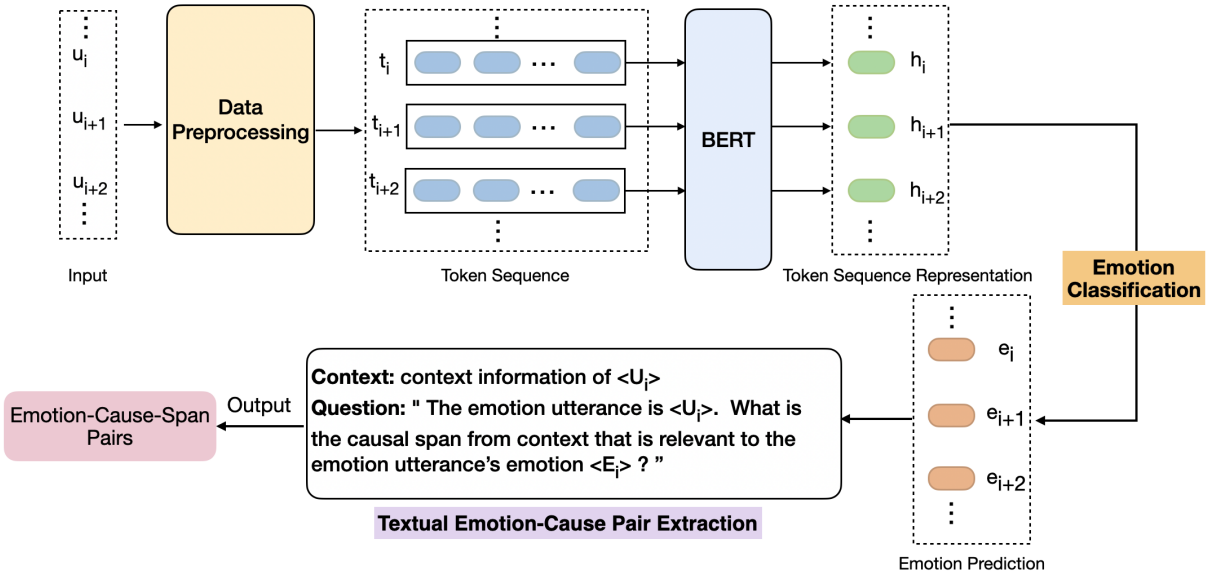


Figure 2: The overall architecture of ECSP consists of three parts: data preprocessing, emotion classification, and textual emotion-cause pair extraction. After preprocessing the origin dataset, BERT is utilized to transform the token sequence with contextual information in the conversation into a semantic representation and predict emotions. Then, extract textual emotion-cause pairs.

Dataset	Train	Val	Test
Positive Samples	7093	900	900
Negative Samples	36778	4247	4247

Table 1: Statistics of the preprocessed dataset, including positive and negative samples.

number of negative examples in which the cause is not expressed in order to better train the model to recognize emotional causes in conversation tasks.

Considering dialogue D and an emotion utterance U_i in D, we construct the complete set of negative examples as $\{U_{Neg}|U_{Neg} \in H(U_i) \setminus C(U_i)\}$, where $H(U_i)$ is the conversational history and $C(U_i)$ is the set of cause utterances for U_i.

Table 1 shows the statistics of the preprocessed dataset.

3.2 Emotion Classification Module

Without the loss of generality, the input can be represented by several utterances, $D = \{U_1, ..., U_i, ...U_n\}$. In our system, BERT is used to build the token sequence representations. Each token sequence is enveloped by predefined special tokens ($[CLS]$, $[SEP]$), $t_i^{'} = \{[CLS], w_{i1}, ..., w_{ik}, [SEP]\}$, where w_{ik} is the k-th token in the i-th utterance's token sequence. The [CLS] token is used for generating representations for classification tasks. The [SEP] token is used to denote the end of a sentence. The utterance's representation h_i is acquired through BERT, which is the final hidden state of [CLS].

$$h_i = BERT(t_i^{'}) \tag{1}$$

The token sequence representation h_i is fed into the Feed-Forward Neural Network (FFNN) layer to obtain the emotion prediction E_i.

$$E_i = Softmax(W^e h_i + b^e) \tag{2}$$

where W^e is a weight and b^e is a bias of the emotion classification layer, respectively.

3.3 Textual Emotion-Cause Pair Extraction Module

In order to implement the extraction of textual span in the ECPE task, we regard this module as a machine reading comprehension (MRC) task. The specific task is defined as follows:

Context: *Context* is the context information $U_j (j \in [1, i])$ of emotion utterance U_i, which is the traversal of all utterances in U_i's conversation history $H(U_i)$.

Question: The *Question* is framed as follows: *" The emotion utterance is $< U_i >$. What is the causal span from the context that causes the emotion $< E_i >$ of the emotion utterance? "*

Answer: The causal span $S \in CS(U_i)$ appearing in U_j if $U_j \in C(U_i)$. For negative examples, S is assigned an empty string.

Among them, emotion utterance U_i is the i-th utterance in dialogue D. $H(Ui)$ is the conversation history set of U_i, a set of all utterances from the beginning of the conversation till the utterance U_i, including U_i. $U_j \in H(U_i)$ is the context of U_i. $C(U_i)$ is the set of cause utterances of U_i, $C(U_i) \in H(U_i)$. $CS(U_i)$ is the cause span set of U_i.

3.4 Loss Function

Loss function is used to evaluate the extent to which the predicted and true values of the model are not the same. For different models and different tasks, the choice of loss function has a great impact on the performance of the model. In this task, the focal loss function is used to better alleviate the problem of unbalanced number of sample categories.

The goal of Focal Loss (Lin et al., 2017) is to address the issue where traditional cross-entropy loss contributes less to the loss of positive samples when there are a large number of easily classified negative samples. The adoption of the focal loss alleviates this issue by balancing the weight assigned to minority classes, facilitating the learning process (Wang et al., 2022).

$$BCEloss(o,t) = -\frac{1}{n}\sum_{i}\Big(t[i]\log(o[i]) + (1-t[i])\log(1-o[i])\Big) \quad (3)$$

As shown in formula 3, we use balance factor to deal with data imbalance in Balance Cross Entropy loss(BCEloss).

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \quad (4)$$

Focal loss reduces the loss weight of easily distinguishable negative samples and increases the dynamic adjustment factor based on BCEloss to achieve the effect of mining difficult samples. We make the model more focused on hard-to-learn samples by setting γ value as 2 in the formula 4, thus the network will not be biased by too many negative examples.

4 Experiments

4.1 Dataset

The SemEval-2024 Task 3 dataset is ECF (Wang et al., 2023a), which contains 1,344 conversations and 13,509 utterances. As shown in Table 2, 55.73% of utterances are labeled with emotion categories, 91.34% of emotions are labeled with corresponding cause, and the same emotion may be caused by multiple cause utterances (the number of emotion-cause pairs is greater than the number of emotion with cause).

Filed	Number
No. of conversations	1,344
No. of utterances	13,509
No. of emotion (utterances)	7,528
No. of emotion (utterances) with cause	6,876
No. of emotion-cause (utterance) pairs	9,272

Table 2: Statistics of ECF dataset.

For each emotion category, the proportion of emotion utterances with reason annotations is shown in Figure 3.

We split the original dataset into 80% train set, 10% valid set, and 10% test set.

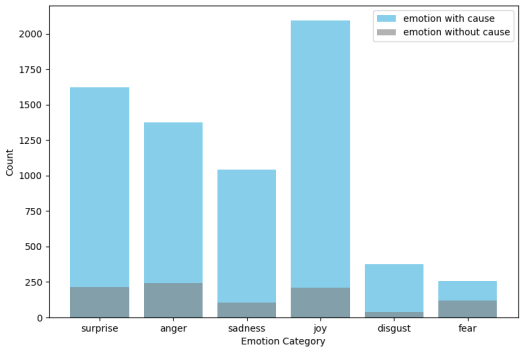


Figure 3: The distribution of emotions (with/ without cause) in different categories.

4.2 Baselines

In our experimental setup, we assume that emotion-cause pairs have two settings:

- Only non-neutral emotional utterances are recognized.
- The cause of emotion only exists in previous or current utterances because speakers cannot predict future utterances in conversational scenarios.

As to the emotion classification module, we used the pre-trained model BERT to obtain the semantic embedding of the input utterance.

BERT: BERT is a deep pre-trained language model based on the Transformer architecture. Devlin et al. (2018) used the Masked Language Model (MLM) to learn rich language representations and achieve SOTA performance in various downstream

Model		Strict			Proportional		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
w/o context	RoBERTa	16.30	12.19	13.57	21.17	17.49	18.42
	SpanBERT	15.03	13.92	13.72	19.33	20.33	18.71
with context	RoBERTa	**18.35**	12.63	14.63	**22.34**	17.51	19.06
	SpanBERT	17.56	**14.41**	**15.18** (3/16)	20.94	**20.20**	**19.63** (7/16)

Table 3: Experimental results of textual emotion-cause pair extraction task. Shown in ( ) is the official ranking.

tasks. In the emotion classification task, we added contextual information to each utterance such that each utterance contains all its previous utterances as context, then used the BERT tokenizer to generate the input tensor of the emotion classification model, encoded it by BERT, and used a linear layer to predict emotions.

As to the textual emotion-cause pair extraction module, we fine-tuned two pre-trained models: RoBERTa and SpanBERT.

RoBERTa: RoBERTa (Liu et al., 2019) is an improved version of the BERT model, adopting more model parameters, more training data, and larger batch sizes. We used a Roberta-base model and added a linear layer on top of the hidden state to calculate the start and end logic of the span.

SpanBERT: SpanBERT (Joshi et al., 2020) is based on BERT, has made specific optimizations in the pre-training stage for the task of predicting spans of text, and has excellent performance in question and answer tasks. We used the SpanBERT-base model fine-tuned on the SQuAD 2.0 dataset as the second baseline model for the textual emotion-cause pair extraction task.

We utilized the PyTorch library (Paszke et al., 2019) and the HuggingFace library (Wolf et al., 2020) on our models and trained and tested them on the Nvidia A800-40G.

4.3 Evaluation Metrics

Since the task of textual emotion-cause pair extraction involves the textual cause span, the organizers of SemEval-2024 Task 3[2] adopted two strategies to determine whether the span is extracted correctly (Wang et al., 2024):

- **Strict Match**: The predicted span should be exactly the same as the annotated span.
- **Proportional Match**: Considering the overlap proportion between the predict span and the annotated one.

[2]https://github.com/NUSTM/SemEval-2024_ECAC

For the **Strict Match**, we firstly evaluate the emotion-cause pairs of each emotion category separately and then further calculate a weighted average of Strict F1-scores across the six emotion categories.

$$\text{Strict}F1 = \sum_{j=1}^{6} w^j \text{Strict}F1^j \tag{5}$$

Where w_j denotes the proportion of the annotated pairs with emotion category j, $j \in \{anger, disgust, fear, joy, sadness, surprise\}$.

For the **Proportional Match**, match each predicted pair with one of the annotated pairs that has the maximum overlap proportion in terms of the cause span (if the predicted span overlaps with multiple annotated spans):

$$overlap_i = \begin{cases} len(ps_i \cap as_k) & [eu_i, ec_i, cu_i] \\ & \text{are correct and} \\ & ps_i \cap as_k \neq \phi, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

$$k = \arg\max_t \frac{len(ps_i \cap as_t)}{len(as_t)} \tag{7}$$

where $len(*)$ denotes the number of textual tokens, ps_i and as_k represent the cause span in the predicted pair pp_i and the annotated pair ap_k respectively. Then the proportional F1-score is calculated based on the overlap length between the predicted span and the annotated span, and a weighted average of the six emotion categories is also calculated.

$$\text{Proportional}F1 = \sum_{j=1}^{6} w^j \text{Proportional}F1^j \tag{8}$$

In the SemEval-2024 Task 3, the organizers initially selected the *Strict F1-score* as the main ranking metric. Due to poor overall results, they eventually switched to using the *Proportional F1-score* as

the main ranking indicator. This also shows that it is very difficult to extract the accurate textual cause span of emotion utterances.

4.4 Results

The experimental results of our work are given in Table 3. As shown in the table, we conducted experiments based on whether to add contextual information to the emotion classification module and gave the performance of two baseline models for the textual emotion-cause extraction task under strict match and proportional match, respectively. Among them, the SpanBERT model using the contextual information emotion prediction module achieved the best performance, with the *Strict F1-score* of 15.18% and the *Proportional F1-score* of 19.63%.

In addition, we draw the following observations:

- Firstly, the context of whole dialogue is curcial for the prediction of causal spans. When contextual information is added to the input utterances in the emotion classification module, the overall performance of the model will be improved to a certain extent. In the RoBERTa model, after adding contextual information, the *Proportional F1-score* increased by 1.36%, and the *Strict F1-score* increased by 1.06%. In the SpanBERT model, the *Proportional F1-score* increased by 0.92%, and the *Strict F1-score* increased by 1.46%.

- Secondly, it can be seen from the experimental results that the SpanBERT model always achieved good performance compared with the RoBERTa model in the textual emotion-cause pair extraction task. When there is no context information in the emotion extraction module, the *Strict F1-score* of the SpanBERT model is 0.12% higher than the RoBERTa model, and the *Proportional F1-score* is 0.29% higher. When there is context information in the emotion extraction module, the *Strict F1-score* of the SpanBERT model is 0.55% higher than the RoBERTa model, and the *Proportional F1-score* is 0.57% higher.

In the official ranking, our team used the three-step learning approach ECSP, which consists of an emotion classification module with contextual information and a textual emotion-cause pair extraction module with SpanBERT as the baseline. The ranking obtained is shown in Table 3. Among them, our team ranked 3rd under the strict match with the *Strict F1-score* of 15.18%, and ranked 7th under the Proportional match with the *Proportional F1-score* of 19.63%.

5 Conclusion

In this paper, we introduce the system implementation of SemEval-2024 Task 3: Textual Emotion-Cause Pair Extraction in Conversations. We propose an integrated system named Emotion-Cause-Span Pair Extraction in Conversation (ECSP), which was implemented in three modules: preprocessing data, emotion classification with contextual information input, and textual emotion-cause pair extraction, and it performed well in the official rankings. In the future, we will utilize this dataset to investigate if the *Speaker* attribute affects the extraction task of emotion-cause pairs, as well as to implement methods such as external knowledge bases to improve our system's recognition performance on ECPE tasks.

Acknowledgment

This paper was partially supported by National Natural Science Foundation of China (Grant number: 61976066), Beijing Natural Science Foundation (Grant number: 4212031), and Research Funds for NSD Construction, University of International Relations (Grant numbers: 2021GA07).

References

Ying Chen, Wenjun Hou, Shoushan Li, Caicong Wu, and Xiaoqiang Zhang. 2020. End-to-end emotion-cause pair extraction with graph convolutional network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 198–207.

Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 179–187.

Zifeng Cheng, Zhiwei Jiang, Yafeng Yin, Na Li, and Qing Gu. 2021. A unified target-oriented sequence-to-sequence model for emotion-cause pair extraction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2779–2791.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. Ecpe-2d: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3161–3170.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, Alexander Gelbukh, and Rada Mihalcea. 2021. Recognizing emotion cause in conversations.

Jingzhe Shan and Min Zhu. 2020. A new component of interactive multi-task network model for emotion-cause pair extraction. In *Journal of Physics: Conference Series*, volume 1693, page 012022. IOP Publishing.

Cheng Wang, Jorge Balazs, György Szarvas, Patrick Ernst, Lahari Poddar, and Pavel Danchenko. 2022. Calibrating imbalanced classifiers with focal loss: An empirical study. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 145–153.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023a. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Fanfan Wang, Jianfei Yu, and Rui Xia. 2023b. Generative emotion cause triplet extraction in conversations with commonsense knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3952–3963.

Penghui Wei, Jiahao Zhao, and Wenji Mao. 2020. Effective inter-clause modeling for end-to-end emotion-cause pair extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3171–3181.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Jun Zhao, Kang Liu, and Liheng Xu. 2016. Sentiment analysis: mining opinions, sentiments, and emotions.

YNU-HPCC at SemEval-2024 Task10: Pre-trained Language Model for Emotion Discovery and Reasoning its Flip in Conversation

Chenyi Liang , Jin Wang and **Xuejie Zhang**
School of Information Science and Engineering
Yunnan University
Kunming, China
liangchenyi@stu.ynu.edu.cn, {wangjin,xjzhang}@ynu.edu.cn

Abstract

This paper describes the application of fine-tuning pre-trained models for SemEval-2024 Task 10: Emotion Discovery and Reasoning its Flip in Conversation (EDiReF), which requires the prediction of emotions for each utterance in a conversation and the identification of sentences where an emotional flip occurs. This model is built on the DeBERTa transformer model and enhanced for emotion detection and flip reasoning in conversations. It employs specific separators for utterance processing and utilizes specific padding to handle variable-length inputs. Methods such as R-drop, back translation, and focal loss are also employed in the training of my model. The model achieved specific results on the competition's official leaderboard. The code of this paper is available at https://github.com/jiaowoobjiuhao/SemEval-2024-task10.

1 Introduction

Navigating the complexities of emotional dynamics within conversations presents a formidable challenge in natural language processing (NLP). Human interactions are characterized by rapid emotional shifts, influenced by context and subtle linguistic nuances, requiring sophisticated models for accurate capture and interpretation. Thus, understanding and precisely identifying emotions, especially within conversations marked by emotional transitions, is a significant and challenging endeavor in NLP research.

The SemEval-2024 competition introduces the Emotion Discovery and Reasoning its Flip in Conversations (EDiReF) task (Kumar et al., 2024), divided into three subtasks designed to explore the nuanced landscape of emotional dynamics within dialogues:

- Subtask 1: Identify and classify the emotional states expressed in each utterance within a conversation (Kumar et al., 2023a). As shown in Table 1, the emotion of each utterance is identified through the first two columns.
- Subtask 2: Identify specific utterances that mark an emotional transition within Hindi-English code-mixed dialogues (Kumar et al., 2022, 2023b). As shown in Table 1, the triggers of emotions are identified through the first three columns.
- Subtask 3: Identify specific utterances that mark an emotional transition within English conversations (Kumar et al., 2022, 2023b). The first three columns in Table 1 identify emotional reversal triggers.

In the previous sentiment analysis work, various hand-crafted features and sentiment lexicons were utilized to construct solution systems. These systems were developed by integrating traditional methods such as Naive Bayes, Support Vector Machines (SVM) (Mohammad et al., 2013), and Decision Trees (Blake, 2007). Following the advent of deep learning, Convolutional Neural Networks (CNNs) (Kim, 2014), based on Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997)architectures, were employed for sentence classification tasks. Additionally, GloVe (Pennington et al., 2014) was utilized for learning sentence features, and Bidirectional Long Short-Term Memory (Bi-LSTM) (Kong et al., 2020; Zhang et al., 2018) models were applied to sentence classification to enhance performance. However, these methods encountered challenges in effectively capturing the contextual information of longer texts. With the progression toward larger models, BERT-based large-scale pre-training models marked a significant breakthrough in sentiment analysis (Zheng et al., 2022)

This study proposes a deep learning system for Task 10 in SemEval-2024. We use a

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 777–784

Speaker	Utterance	Emotion	Trigger
Sp1	I had an awful day today!	Sad	0
Sp2	Oh no! What happened?	Sad	0
Sp1	Somebody ate my sandwich!	Sad	0
Sp2	I can make you a new one right now!	Joy	1
Sp1	That would be great! Thanks!	Joy	0

Table 1: Examples of EDiReF

decoding-enhanced bert with disentangled attention(DeBERTa) (He et al., 2020)sequence classification model as the base model. Our enhancement to the DeBERTa model introduces a pivotal integration of specialized mechanisms for processing [SEP] tokens and handling label padding with -1, along with the innovative incorporation of a KL divergence (Wu et al., 2021) loss function, known as R-drop. This strategic amalgamation ensures that each utterance within a conversation is precisely mapped to its corresponding emotional state, facilitating a more accurate representation of emotional dynamics. Introducing R-drop is critical in preventing overfitting by enforcing consistency between the model's outputs for various data sub-samples, thus enhancing the model's generalization ability across different conversational contexts. The contributions of this study are as follows.

- We introduce a foundational model utilizing a pre-trained Deberta sequence classification model for the label sequence classification issue.
- Incorporation of KL Divergence Loss (R-drop) for Overfitting Prevention and adoption of focal loss to address data imbalance issues.
- The model employs [SEP] tokens and -1 padding to align utterances with their corresponding labels and grasp the context within conversations.

The remainder of this paper is organized as follows: Section 2 provides an overview of our proposed model and system. Section 3 conducted the experiments to analyze the effectiveness of the proposed method. The paper concludes with a summary and reflections in Section 4.

2 System Description

This section delves into the architecture of the proposed model, detailing its essential components:

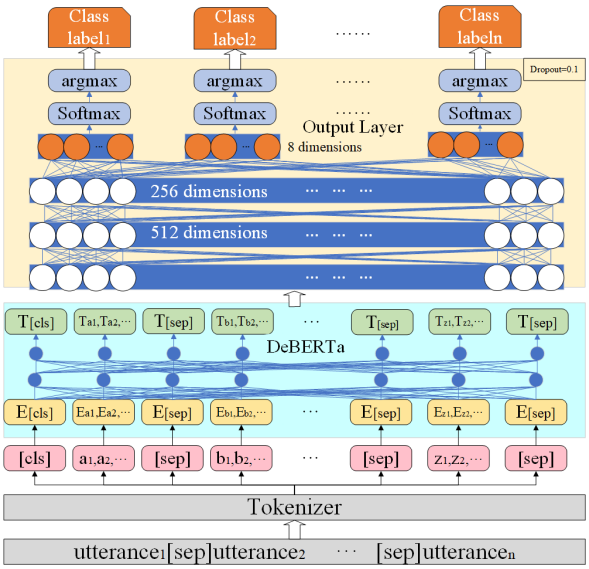


Figure 1: Multi-emotion label sequence classification model

the tokenizer, the pre-trained Deberta model, and the implementation of Regularized Dropout and Focal loss for Neural Networks. Specifically, the model tailored for Task 1, which addresses the multi-label sequence classification problem, is illustrated in Figure 1. Meanwhile, the models designed for Tasks 2 and 3, focusing on binary sequence classification issues, are depicted in Figure 2.

2.1 Tokenizer

Transforming raw text into a machine-readable format is a preliminary step for many NLP tasks. To achieve this, a tokenizer is utilized, segmenting the text into discrete elements and encoding them uniquely. In our model, the DeBERTa tokenizer, mainly designed for handling long texts in sequence classification challenges, is employed to process the text for NLP tasks. Input texts are segmented to accommodate the extensive length of dialogues in subtasks 1 and 2 using a 2048 token cutoff, ensuring comprehensive coverage of conversa-

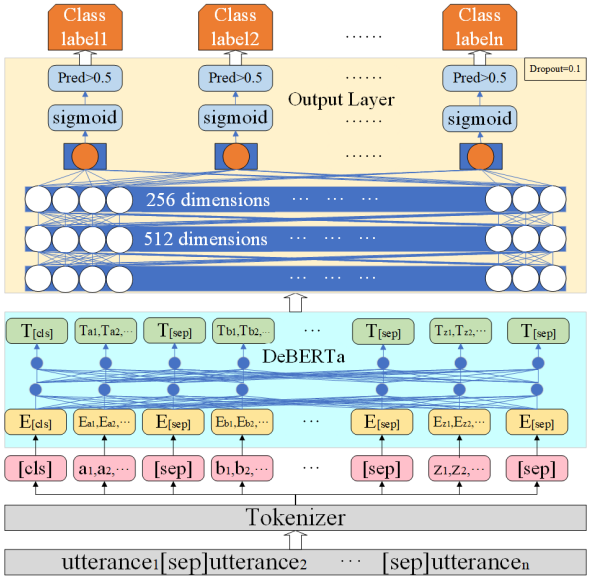


Figure 2: Binary label sequence classification model

tions without truncating critical emotional context in later utterances. For subtask 3, a 1024 token limit is applied, optimizing for shorter textual inputs. The final output X of the tokenizer is denoted as:

$$X = [CLS]\, a_1 \cdots a_n\, [SEP]\, b_1 \cdots b_m\, [SEP] \cdots z_1 \cdots z_p\, [SEP] \tag{1}$$

where n, m, and p denote the lengths of distinct utterances within the dialogue. With [CLS] marking the start and [SEP] serving as a delimiter between utterances, it ensures the model recognizes dialogue flow. For subtasks 1 and 2, sequences shorter than 2048 tokens are padded with zeros, and longer ones are truncated to maintain this limit, optimizing for more extended dialogues. Conversely, subtask 3 employs a 1024 token threshold, adjusting for its specific data structure and requirements.

2.2 DeBERTa Model

DeBERTa enhances BERT's (Devlin et al., 2019) architecture by introducing disentangled attention and an enhanced mask decoder, making it highly suitable for complex dialogue tasks requiring a detailed understanding of context and word positions. Like BERT, DeBERTa comprises two core components: an Embedding block for initial word vector representation and a Transformer Encoder block for deep contextual processing. Additionally, DeBERTa introduces a third major component, the Enhanced Mask Decoder (EMD).

Following the tokenizer's segmentation of input texts and incorporation of special tokens ([CLS] and [SEP]), these tokens are embedded, capturing the nuances of words as vectors that signify their meanings and relationships. The Transformer Encoder further processes these vectors by employing disentangled attention to analyze dialogues' contextual relationships and depth intricately. The EMD, leveraging content and positional information, refines the model's ability to predict and understand masked language elements, thoroughly comprehending dialogue intricacies. Consequently, the final layer's hidden state representation, denoted as H_L, is passed to the output layer, where L represents the number of layers in the Transformer.

2.3 Output Layer

Subtask1. This subtask involves multi-sequence sentiment classification, with the model designed to recognize [sep] and label padding of -1. This setup allows for processing the DeBERTa model's sequence output through a custom classifier, generating logits for each utterance to predict labels, detailed in section 3. The layer initially maps the data dimensions from L to 512 dimensions, then applies the ReLU activation function and dropout to refine and classify the data further, followed by mapping from 512 to 256 dimensions, adding ReLU and Dropout again, and finally mapping to the label dimension (8 dimensions) to obtain logits. After obtaining the classification probability distribution P , calculate the loss with the real classification label y and learn the model weight. The calculation formula of the probability distribution is as follows.

$$P = softmax\left(W_0 H_0 + b_0\right) \tag{2}$$

where W_0 is the weight matrix of the final linear layer, with dimensions of $R^{8\times256}$, H_0 is the feature vector input to this linear layer, with a dimensionality of 256; and b_0 is the bias term, with a dimensionality of 8.

Subtask2&subtask3. These two subtasks involve binary sequence classification, where the main difference in the output layer from subtask 1 is the transformation of the model's output logits into probabilistic classifications through a sigmoid layer. The calculation formula of the probability distribution is as follows.

$$P = sigmoid\left(W_1 H_1 + b_1\right) \tag{3}$$

where W_1 is the weight matrix of the final linear layer, with dimensions of $R^{1\times256}$, H_1 is the feature vector input to this linear layer, with a dimensionality of 256, and b_1 is the bias term, with a dimensionality of 8.

2.4 Methods

Regularized Dropout. Due to the existence of dropout, the same model with identical inputs will produce two distinct distributions, effectively treating them as two different network models. Denoted as $P_\theta(y|x)$ and $P_\theta^{'}(y|x)$, these distributions represent the output probabilities of the model under dropout conditions. The primary objective of R-Drop is to minimize the KL Divergence between these two distributions throughout the training process. Given the asymmetry of KL divergence, a globally symmetric version is indirectly employed by interchanging the positions of these distributions, a concept known as bidirectional KL divergence. Furthermore, the model is trained on both distributions' negative log-likelihood (NLL) loss terms. Given (x_i, y_i) as training set input ,The final loss is as follows:

$$\begin{aligned} L_{KL}^i &= \alpha \Big[D_{KL}\left(P_\theta(y_i|x_i)||P_\theta^{'}(y_i|x_i)\right) \\ &\quad + D_{KL}\left(P_\theta^{'}(y_i|x_i)||P_\theta(y_i|x_i)\right)\Big] \\ L_{NLL}^i &= -\log P_\theta(y_i|x_i) - \log P_\theta^{'}(y_i|x_i) \\ L_{R-drop}^i &= L_{KL} + L_{NLL} \end{aligned} \tag{4}$$

Focal Loss. Focal Loss (Lin et al., 2017) is utilized in our model as the primary loss function, specifically designed to mitigate the impact of class imbalance by dynamically adjusting the importance of each class and the difficulty of each sample. Two parameters α and γ are introduced to modulate each class's loss contribution and focus more on challenging, misclassified samples rather than those easily classified. P_t represents the probability of class t output by softmax or sigmoid function and α_t is a training parameter. The formula is listed as follows.

$$FL\left(P_t\right) = -\alpha_t(1-P_t)^\gamma \log\left(P_t\right) \tag{5}$$

3 Experimental Results

3.1 Datasets

The training sets for these three subtasks are derived from dialogues in various scenarios within TV dramas. In subtask 1, 343 training sets are provided, including four columns: episode, speakers, emotions, and utterances, with eight types of emotions contained within the emotions column. For subtasks 2 and 3, 4893 and 4000 training sets are provided, each with an additional column named triggers compared to subtask 1.

3.2 Evaluation Metrics

The evaluation tools employed for these three subtasks are Precision, Recall, and Micro-F1, with their formulas categorized as follows:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ F1 &= \frac{2\times Precision \times Recall}{Precision + Recall} \end{aligned} \tag{6}$$

3.3 Implementation Details

Training Set Preprocessing. To align each utterance with its label throughout an entire dialogue and to learn the relationships within the dialogue, each dialogue is separated by [sep]. The label data are filled with -1 to match the maximum number of utterances in the training and validation sets, and a mask is incorporated into the model. This approach ensures that labels marked as -1 are excluded from loss calculation, allowing the model to handle dialogues of varying lengths. Specifically, the maximum number of utterances for subtasks 1 and 2 is 106, while for subtask 3, it is 24. The labels for subtask 1 are emotions, with eight types: anger, contempt, disgust, fear, joy, neutral, sadness, and surprise. These are mapped to data values 0-7, facilitating correct processing by the model. For subtasks 2 and 3, initially in string format as 0 and 1, the label data are converted to floating-point numbers 0.0 and 1.0. Given the limited training dataset for subtask 1, data cleaning and normalization are first performed using ekphrasis, which improves the model's learning from dialogues. Text augmentation is then conducted through back-translation (Edunov et al., 2018) and synonym replacement; Hindi dialogues are translated into English and then back, while the process is reversed for English dialogues. Synonym replacement involves exchanging words with the same meaning for different expressions. Finally, subtask 1 is expanded to 1029 training sets.

Imbalanced Data Handling. Due to the predominant proportion of neutral and joy labels in subtask

contempt	542
disgust	127
fear	514
joy	1596
neutral	3909
sadness	558
surprise	441

Table 2: Occurrences of Emotional Labels in Subtask

1, as illustrated by the quantities in Table 2, as well as the prevalence of the 0.0 label in subtasks 2 and 3, focal loss and oversampling methods (Chawla et al., 2002) have been utilized. This approach enables the model to learn more effectively from samples that appear less frequently, thereby enhancing the model's performance.

Prediction Challenges. When tokenizing text inputs, lengths of 2048 were selected for truncation in subtasks 1 and 2, while 1024 was chosen for subtask 3. However, during the prediction phase for subtask 2, the number of labels predicted fell short of the expected count. This shortfall could be attributed to dialogues in the test set that exceed the maximum length of 2048. The constraints posed by GPU capabilities also resulted in our model's inability to fully perform the prediction task for subtask 2. We hope to try using Longformer (Beltagy et al., 2020) to address the issue of long dialogues in the future.

Model Comparison. For all tasks, `bert-base-cased`, `bert-large-cased`, `deberta-base`, and `debertav2-xlarge` were compared. When employing the `debertav2-xlarge` model, the AdaLoRA (Zhang et al., 2023) model was used for fine-tuning to prevent exceeding the GPU memory limits.

Optimizer and Loss Parameter Configuration. The AdamW (Loshchilov and Hutter, 2017) optimization was employed to train the model across all subtasks, with a batch size of 1. To achieve the expected results, we experimented with different learning rates and epochs to observe their impact on the F1 score for Subtask 1 using the deberta-base model. Figures 3 and 4 are presented below. For subtask 1, the learning rate for AdamW was set at 5e-6, while for subtasks 2 and 3, it was established at 5e-5. Focal loss parameters for subtask 1 were defined as alpha=0.1 and gamma=0.3, while for subtasks 2 and 3, the parameters were set to

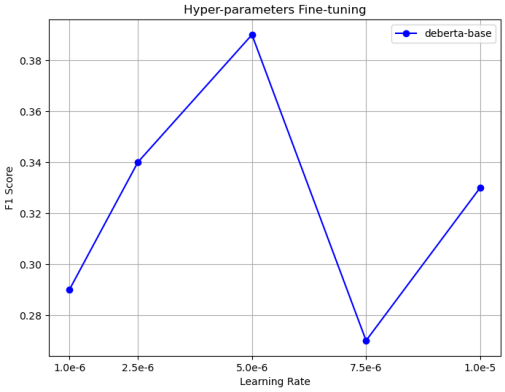


Figure 3: The impact of different learning rates on the F1 score for Subtask 1

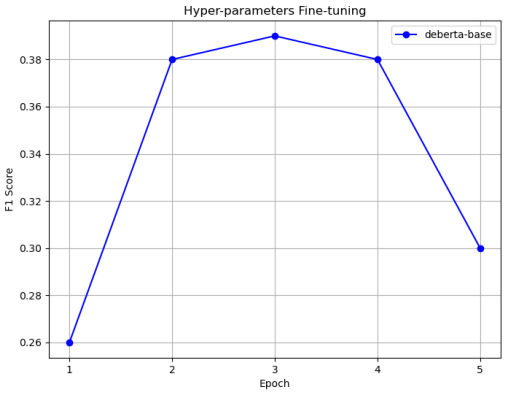


Figure 4: The impact of different learning rates on the F1 score for Subtask 1

alpha=1 and gamma=5.

3.4 Results and Analysis

Subtask1. Validation set results for different models for the multi-label sequence classification task are presented in Table 4. Performance increases from the bert to the DeBERTa phase, yet a significant decline occurs at the `debertav2-xlarge` model phase. This decline may be attributed to the large parameter size of the `debertav2-xlarge` model and the small dataset size, making it challenging for the model to learn features from a small dataset. The overall low scores for the model could be due to the approach of predicting the entire dialogue segment and calculating loss against actual values rather than calculating loss for each utterance individually. This approach might have contributed to the suboptimal performance of our model. Another potential reason could be the selection of 2048 as the trunca-

Speaker	Utterance	Predicted label	True label
maya	indu tumne vah mere earplugs dekhe hain	neutral	neutral
indravardhan	earplugs kyon	anger	surprise
maya	<time>baj rahe hain na madhubhai ki bhatiji ka sone ka time ho gaya hai	neutral	neutral
indravardhan	are baap re yyane announcement shuru ho jayegi	anger	fear
dvd player	train sound	anger	neutral
maya	a <elongated>	anger	fear

Table 3: Model's prediction results on the test set for Subtask 1

Model	Dev set		
	P	R	F1
DeBERTa-base	0.39	0.39	0.39
DeBERTaV2-xlarge	0.18	0.18	0.18
Bert-base	0.28	0.28	0.28
Bert-large	0.31	0.31	0.31

Table 4: Validation set results for different models for Subtask 1

Model	Dev set		
	P	R	F1
DeBERTa	0.25	0.25	0.25
DeBERTa-focalloss	0.36	0.36	0.36
DeBERTa-rdrop	0.35	0.35	0.35
DeBERTa-focalloss-rdrop	0.39	0.39	0.39

Table 5: Validation set results for different methods for Subtask 1

Model	Dev set		
	P	R	F1
DeBERTaV2-xlarge	0.90	0.90	0.90

Table 6: Validation Set Results for Subtask 2

tion value. Although this ensures that a few longer dialogue texts are fully captured, it may hinder the model's ability to learn information from long-distance texts for most shorter dialogues, leading to poor learning outcomes. There is a keen interest in attempting to segment longer texts in the future to mitigate the adverse effects on learning caused by long texts.

As indicated, the model `deberta-base` outperforms others on the validation set. Subsequent experiments will explore the impact of different methods on the model's performance based on `deberta-base`. The results are presented in Table 5, which reveals that the baseline model, not utilizing focal loss or r-drop, and instead using CrossEntropyLoss as the loss function, achieves an F1 score of only 0.25. Introducing either focal loss or r-drop results in improved scores, reaching 0.36 and 0.35, respectively. Combining these two methods and applying them to the `deberta-base` model on the validation set increases the F1 score to 0.39, outperforming the previous three configurations. The experiments demonstrate that both rdrop and focal loss contribute to enhancements in model performance.

The model `deberta-base-focalloss-rdrop` was employed to make predictions on the test set, with the results presented in Table 7, which indicates that the predictions for shorter sentences are not very accurate, which may be due to the model's insufficient learning of brief phrases. Another reason could be that the pre-trained model, `deberta-base`, was primarily trained in English, resulting in inadequate learning for languages like Hindi. Applying a multilingual model might yield better results, and further experiments are hoped to be conducted.

Subtask2. Validation Set Results for the Binary Label Sequence Task are shown in Table 6.When the `debertav2-xlarge` model was attempted for prediction, 2048 was selected as the truncation length for tokenizing the test set dialogues. It was found that the number of predicted labels did not meet the expected count, possibly due to dialogues exceeding the length of 2048, leading to this shortfall. Given the GPU constraints, our model could not effectively predict the test set.

Subtask3. Validation Set Results are presented in Table 8.It was observed that the values of precision and recall are identical across all tasks, which

Speaker	Utterance	Emotion	Predicted trigger	True trigger
Mark	why do all your coffee mugs have numbers on the bottom	surprise	0.0	0.0
Rachel	oh. that is so Monica can keep track. That way if one of them is missing, she can be like, where is number <number>?! <repeated>	anger	0.0	0.0
Rachel	y ' know what ?	neutral	0.0	0.0

Table 7: Model's prediction results on the test set for Subtask 3

Model	Dev set		
	P	R	F1
DeBERTa-base	0.82	0.82	0.82
DeBERTaV2-xlarge	0.82	0.82	0.82
Bert-base	0.82	0.82	0.82
Bert-large	0.82	0.82	0.82

Table 8: Validation set results for different models for Subtask 3

may be attributed to using micro-F1 as the evaluation metric and calculating loss based on entire dialogue segments rather than extracting individual utterances. This approach resulted in identical calculated values. The prevalence of 0.0 labels in every dialogue segment possibly made it challenging for the model to learn and perform well on the test set. The aspiration is to learn more practical models in the future to address this issue. The results obtained from predicting the test set using the `debertav2-xlarge` model are shown in Table 7, which shows that the model performs well in identifying non-emotional triggers. Based on my overall prediction results, the model's ability to predict triggers is unsatisfactory, which is an area I should aim to improve in the future.

4 Conclusions

This paper proposes a deep learning model for sentence sequence classification tasks, utilizing the DeBERTa sentence sequence classification model as the foundation. Achievements have been made in the final submission for SemEval-2024 Task10. However, there remains significant room for improvement in both the model and its parameters. Therefore, in future studies, enhancements will be made to the model to achieve better results.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos.61966038 and 62266051.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Catherine Blake. 2007. The role of sentence structure in recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 101–106.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

S Hochreiter and J Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014*

Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1746–1751. ACL.

Jun Kong, Jin Wang, and Xuejie Zhang. 2020. Hpcc-ynu at semeval-2020 task 9: A bilingual vector gating mechanism for sentiment analysis of code-mixed text. *arXiv preprint arXiv:2010.04935*.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Md Shad Akhtar, Tanmoy Chakraborty, et al. 2023a. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. *arXiv preprint arXiv:2310.13080*.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023b. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.

You Zhang, Jin Wang, and Xuejie Zhang. 2018. Ynu-hpcc at semeval-2018 task 1: Bilstm with attention based sentiment analysis for affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 273–278.

Guangmin Zheng, Jin Wang, and Xuejie Zhang. 2022. Ynu-hpcc at semeval-2022 task 6: Transformer-based model for intended sarcasm detection in english and arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 956–961.

YNU-HPCC at SemEval-2024 Task 2: Applying DeBERTa-v3-large to Safe Biomedical Natural Language Inference for Clinical Trials

Rengui Zhang, Jin Wang and **Xuejie Zhang**
School of Information Science and Engineering
Yunnan University
Kunming, China
zrg@mail.ynu.edu.cn, {wangjin,xjzhang}@ynu.edu.cn

Abstract

This paper describes the system for the YNU-HPCC team for SemEval2024 Task 2, focusing on Safe Biomedical Natural Language Inference for Clinical Trials. The core challenge of this task lies in discerning the textual entailment relationship between Clinical Trial Reports (CTR) and statements annotated by expert annotators, including the necessity to infer the relationships in texts subjected to semantic interventions accurately. Our approach leverages a fine-tuned DeBERTa-v3-large model augmented with supervised contrastive learning and back-translation techniques. Supervised contrastive learning aims to bolster classification ac-curacy while back-translation enriches the diversity and quality of our training corpus. Our method achieves a decent F1 score. However, the results also indicate a need for further en-hancements in the system's capacity for deep semantic comprehension, highlighting areas for future refinement. The code of this paper is available at: https://github.com/RGTnuw/RG_YNU-HPCC-at-Semeval2024-Task2.

1 Introduction

Clinical trials constitute a critical component of medical research, evaluating the safety and efficacy of new treatment methods, medications, or medical devices (Avis et al., 2006). A significant num-ber of Clinical Trial Reports (CTRs) are generated throughout clinical trials. These reports typically encompass information on research design, patient demographics, treatment protocols, outcomes (such as response rates and side effects), and overall conclusions. Such comprehensive and transparent reporting of trial results provides the scientific community and the public with valuable information, informing future research and clinical practice (Zhang et al., 2020). However, the challenge is compounded by over 400,000 Clinical Trial Reports (CTRs) and their rapidly accelerating publication rate. Conducting a comprehensive review of all pertinent literature when devising treatments is impractical (DeYoung et al., 2020).

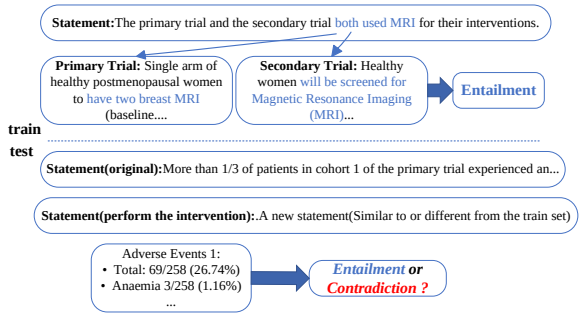


Figure 1: demonstrates textual entailment and contradiction between the medical statements and clinical trial records. Add interventions in the development and test sets.

In response to this challenge, Natural Language Inference (NLI) (Bowman et al., 2015; Devlin et al., 2019) presents a viable approach for the extensive interpretation and retrieval of medical evidence, facilitating enhanced precision and efficiency in personalized evidence-based care (Sutton et al., 2020). This task (Jullien et al., 2024) delineates the objective as classifying the inferential relationship between one or two CTR premises and a statement as either entailment or contradiction. Various interventions were applied to statements in the test and development sets, preserving or inverting entailment relations. It is imperative to ensure that inferred outcomes are justified, i.e., make correct predictions for the right reasons, and identical semantics yield consistent results, as shown in Figure 1 .

In the previous task (Jullien et al., 2023b), large language models (LLM) have achieved commendable performance(Zhou et al., 2023; Vladika and Matthes, 2023). However, the model's performance must improve when facing numerical reasoning, abbreviation, and other problems. DeBERTa-v3-large (He et al., 2023) maintained competitiveness

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 785–791

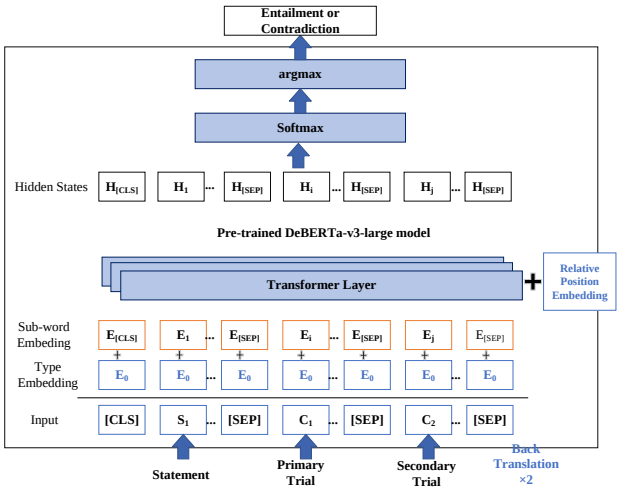


Figure 2: The structure of the system

with leading generative approaches, demonstrating that enhancements in model size correlate with performance improvements. Specifically, augmenting the model's scale directly boosts performance, significantly surpassing the gains from biomedical pre-training. Thus, validating the development set, we opted to submit results with higher experimental scores. Our approach involved fine-tuning the pre-trained DeBERTa-v3-large model, supplemented with supervised contrastive learning and back-translation techniques.

Comprehensive experiments showed that our system achieved a maximum F1 score of 0.77, securing the seventh position on the leaderboard. However, the model exhibited suboptimal performance in faithfulness and consistency metrics, indicating a weaker predictive capacity for data altered by interventions, highlighting areas for future enhancement.

The remainder of this paper is organized as follows. Section 2 describes the model and method used in our system, Section 3 discusses the results of the experiments, and finally, the conclusions are drawn in Section 4.

2 System Description

This section will describe the architecture of the proposed model in detail, including the data loader and back translation, the pre-trained model DeBERTa-v3-large, and supervised contrastive learning; the system model we proposed is shown in Figure 2.

2.1 Data preprocessing

Before feeding statements and CTRs into the model, preprocessing is performed. Initially, data augmentation is conducted through back-translation, a widely adopted technique involving translating text into another language and then back to the original language. This process, achieved via automatic translation systems, utilized Baidu's machine translation API[1] in this study, effectively doubling the training data. Given training data $\mathcal{D} = \{S, C, y\}$, y is the corresponding ground-true label, S is the medical hypothesis sentences, C is the corresponding CTR of the sentence, data loader is applied to transform training data as:

$$X = [CLS]s_1 s_2 \dots s_n[SEP]c_1 c_2 \dots c_m[SEP] \quad (1)$$

where s is the hypotheses with length n and c denotes the CTR reports with length m. [CLS] is a special mark indicating the beginning of the text sequence; [SEP] indicates the separator between text sequences. A similar process compares two CTRs, appending [SEP] and concluding similarly. Sequences exceeding 512 tokens are truncated, while shorter ones are padded.

2.2 Pre-trained DeBERTa-v3-large model

Given the commendable performance exhibited by the DeBERTa-v3 model (He et al., 2023) on this task (Jullien et al., 2023b) and the positive correlation between model parameter size and performance, the DeBERTa-v3-large model was selected as the baseline. Furthermore, an exploration was conducted with several DeBERTa-v3-large models fine-tuned on other NLI datasets available on the Hugging face[2] (Sileo, 2023; Laurer et al., 2023). The pre-trained datasets include MultiNLI, FeverNLI, ANLI, LingNLI, and WANLI. The DeBERTa-v3-large model has 24 layers and a hidden size of 1024. It has 304M backbone parameters with a vocabulary containing 128K tokens, which introduces 131M parameters in the Embedding layer.DeBERTa encodes the input text into the logits,

$$\mathbf{H} = \mathrm{Enc}(X; \theta) \quad (2)$$

where $\mathbf{H} \in \mathbb{R}^d$ is the logits with the dimensionality of d. The [CLS] token, positioned at the beginning of the input sequence, yields a hidden representation $\mathbf{H}_0$, signifying the sequence's initial contex-

[1] https://api.fanyi.baidu.com/
[2] https://huggingface.co/

tual semantic feature within the vector H. Following the acquisition of Following the acquisition of $\mathbf{H}_0$ the [CLS] representation, a fully connected layer leverages it to predict the corresponding label for the input text. The output is a softmax function,

$$\hat{\mathbf{y}} = \text{softmax}(W^0\mathbf{H}_0 + (h^0) \tag{3}$$

where $W^0 \in R^{d\times k}$ represents the weight of the fully connected layer, h^0 represents the offset of the fully connected layer, and k represents the number of classification labels.

2.3 Supervised Contrastive Learning Loss

Contrastive learning (Khosla et al., 2020)is a technique that learns to embed representations of similar samples closer together in the embedding space while pushing apart representations of dissimilar samples. In our model training, we employed this approach by incorporating a supervised contrastive loss alongside the cross-entropy loss. We hypothesized that this method would effectively handle interventions because it encourages the model to learn invariant features across different variations of the data introduced by such interventions (Feng et al., 2023). This invariance is critical for the model to generalize well to new, unseen data that might contain similar variations. Furthermore, we experimented with the R-drop technique R-drop(liang et al., 2021) to further enhance the model's generalization capabilities. However, results from Section 3 suggest that our implementation did not yield the expected improvements. This could be attributed to suboptimal parameter settings or the specific characteristics of our dataset and model size, which might have led to underfitting.

The cross-entropy loss is employed to guide the model towards accurate classification, which measures the discrepancy between the probability distribution predicted by the model and the actual distribution of the proper labels. The contrastive loss part h_i represents a feature vector, and h_{i+} is another feature vector within the same category. The dot product operation effectively calculates the cosine similarity between normalized feature vectors, τ which is the temperature parameter that modulates the model's ability to differentiate between pairs of samples. As the temperature parameter increases, the contrastive loss tends towards treating all sample pairs equally. In contrast, decreasing the temperature parameter focuses the model's attention on the most challenging negative samples. The indicator function ensures that a sample is not com-pared with itself. The SCL loss aims to bring samples of the same category closer together while pushing samples from different categories apart, thereby enhancing the discriminative power of the features. α and β hyperparameters are used to balance the contribution of each loss component. Ultimately, we formulated our loss function as follows to combine both losses effectively:

$$L_{CE} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{4}$$

$$L_{SCL} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(h_i \cdot h_{i+}/\tau)}{\sum_{j=1}^{N} 1_{[j+i]} \exp(h_i \cdot h_j/\tau)} \tag{5}$$

$$L = \alpha L_{CE} + \beta L_{SCL} \tag{6}$$

3 Experimental Results

Datasets. NLI4CT (Jullien et al., 2023a) is designed to assist in developing and benchmarking models for clinical NLI. Which consists of annotated Clinical Trial Reports (CTRs) focused on breast cancer research. Each CTR is meticulously structured into four key sections: (1)Eligibility Criteria: Specifies the prerequisites for patient inclusion in the clinical trial, detailing necessary conditions and characteristics. (2)Intervention: Describes the treatment regimen, including type, dosage, frequency, and duration of the administered treatments. (3)Results: Reports on the trial's participant count, outcome measures, metrics, and findings. (4)Adverse Events: Documents observed signs, symptoms, and any adverse effects encountered by patients during the clinical trial's course. The premises for NLI4CT are sourced from 1,000 publicly accessible Breast Cancer Clinical Trial Reports (CTRs) in English, published on ClinicalTrials.gov[3]. There are 999 breast cancer CTRs in the dataset. The datasets, which are divided into train, development, and test sets, contain a total of 2400 annotated statements. The distribution of labels be-tween the train and development sets is even. Upon employing back-translation, the volume of training data was effectively doubled. Notwithstanding, the test dataset substantially exceeds the size of the training dataset, a scenario that underscores the critical need for models to exhibit

[3]https://clinicaltrials.gov/

Class	Training	Validation	Enhancement Training	Test
Contradiction	850	100	1700	
Entailment	850	100	1700	
Total	1700	200	3400	5667

Table 1: Data distribution

robust generalization capabilities. The distribution of the dacctaset is shown in Table 1.

Evaluation Metrics. The task has three metrics; the **Macro F1-score** is a foundational metric, offering a balanced measure of precision and recall across the dataset's categorial spectrum without any semantic interventions. **Faithfulness** quantifies a model's capacity to adjust its predictions for the right reasons, especially when confronted with semantic-altering interventions. This metric illuminates a model's understanding of the underlying semantics, rewarding models that exhibit agile adaptability to semantic nuances. **Consistency** gauges a model's reliability in producing uniform outputs for semantically equivalent stimuli, regardless of the correctness of the final prediction. This metric champions models that demonstrate robustness in semantic representation, ensuring that semantically similar inputs yield consistent predictions. The formula for the three indices is expressed as follows:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

$$Faithfulness = \frac{1}{N} \sum_{i=1}^{N} |f(y_i) - f(x_i)| \quad (8)$$

where $x_i \in C$ with $\text{Label}(x_i) \neq \text{Label}(y_i)$ and $f(y_i) = \text{Label}(y_i)$.

$$Consistency = \frac{1}{N} \sum_{i=1}^{N} 1 - |f(y_i) - f(x_i)| \quad (9)$$

where $x_i \in C$ where $\text{Label}(x_i) = \text{Label}(y_i)$, N is the total number of sentences, x_i and $y_i$denote the modified and original statements, respectively. The F1 score primarily aims to evaluate the model's performance on data without interventions. At the same time, the other two metrics assess the ability to make correct judgments post-intervention, indicating the model's deeper and more logical understanding of semantic information.

Implementation Details. All compared models were downloaded from HuggingFace. We fine-tune these models on the training set. The models

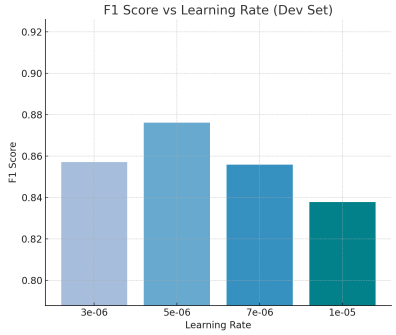


Figure 3: F1 scores on the development set for different learning rates, using the same pre-trained model and other parameters

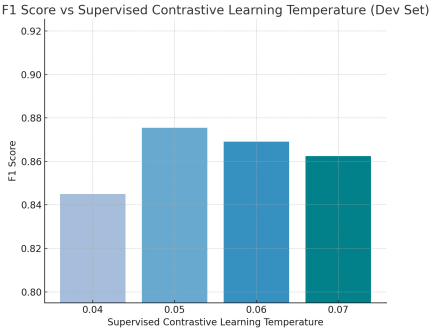


Figure 4: F1 scores on the development set for Supervised Contrastive Learning Temperature, using the same pre-trained model and other parameters

are evaluated on the validation every ten steps using precision, recall, and F1 scores. An Adam optimizer (Loshchilov and Hutter, 2019) updates the param-eters. The warmup strategy (He et al., 2016)is used to optimize the learning rate, and a fixed ran-dom seed is used.

Parameters Fine-tuning. Initially, manual adjustments were made to the hyperparameters, including the learning rate and the temperature for the contrastive loss function. Due to constraints imposed by GPU memory capacity, the batch size for training data was fixed at 4, with results illustrated in Figures 3 and 4. Upon identifying the approximate range of optimal parameters, the Optuna framework (Akiba et al., 2019) was employed for hyperparameter tuning. The parameters yielding the highest F1 score on the development set were selected for further training and model saving. The inference results were then uploaded to the platform.

Comparative Results and Discussion. Table 2 demonstrates that models pre-trained on additional datasets surpass the baseline model in performance on the development set. Nonetheless, it is shown

Model	Pre-training data	Loss	F1
Deberta-v3-large		CE	0.8018
Deberta-v3-large	600+ tasks	CE	0.8518
Deberta-v3-large	MultiNLI+FeverNLI+ANLI+LingNLI+WANLI	CE	0.8504
Deberta-v3-large	MultiNLI+FeverNLI+ANLI+LingNLI+WANLI+Other classification tasks	CE	0.8173
Deberta-v3-large	600+ tasks	CE+R-drop	0.8487
Deberta-v3-large	600+ tasks	CE+SCL	0.8544
Deberta-v3-large +Back Translation	600+ tasks	CE+SCL	0.8625
Deberta-v3-large +Back Translation	MultiNLI+FeverNLI+ANLI+LingNLI+WANLI	CE+SCL	0.8834
Deberta-v3-large +Back Translation	MultiNLI+FeverNLI+ANLI+LingNLI+WANLI+Other classification tasks	CE+SCL	0.8755

Table 2: Comparative results of experiments in the dev set

F1(dev)	F1(test)	Faithfulness	Consistency
0.8755	0.77	0.67	0.72
0.8834	0.75	0.73	0.74

Table 3: Optimal results of the test

that an excess of pre-training tasks yields minimal enhancements in model performance, such as the model that was fine-tuned with multi-task learning across over 600 tasks from the task source collection (Sileo, 2023; Laurer et al., 2023). It was also observed that R-drop might not be ideally suited for this task, potentially due to suboptimal parameter selection. It can be seen from Figure 3 and Figure 4 that the learning rate of the model is more suitable in the vicinity of 5e-6.

In contrast, the temperature of comparative learning is difficult to control, and the model performance is not linear, which needs further exploration. A degree of performance improvement was achieved through supervised contrastive learning. The highest F1 score of 0.8834 on the development set was achieved by combining supervised contrastive learning with the back-translation method. However, an F1 score of 0.75 was only achieved by this model on the test set, equating to the score of 11th place. Scores of 0.73 and 0.74 were reached on the other two metrics, comparable to the scores of the 17th and 9th places, respectively. Despite this, only the highest F1 scores are listed on the leaderboard. Another model of ours reached an F1 score of 0.77, placing it 9th, yet the scores on the other two metrics were not as high, placing 17th and 13th, respectively.

Such scores suggest that predictions are often not based on valid reasoning by the model. Accurate conclusions, when reached, may be derived from incorrect premises or misinterpretations of the input data, suggesting an insensitivity to semantic changes or an incapacity to reflect these changes accurately in its predictions. The reduction of this score indicates the model's prediction instability in the absence of significant semantic alterations, reflecting an excessive sensitivity to minor variations in input or a failure to capture and maintain the input's core semantic features accurately.

These findings reveal deficiencies in our system's ability to understand and process complex and subtle semantic changes despite adequate performance, as indicated by the F1 score. An overreliance on specific data distributions, a lack of generalizability, or challenges in explaining decisions in practical applications may result. To improve the model's Faithfulness and Consistency, it may be necessary for further research and improvements to be conducted on the model's internal representations and training processes or for additional mechanisms to be integrated for better processing of semantic information.

4 Conclusion

This paper introduces a system based on fine-tuning and pre-training Deberta-v3-large for SemEval2024 task 2, targeting safe biomedical NLI for clinical trials. Achieving seventh out of 32 with an F1 of 0.77 showcases the effectiveness of multi-task pre-training, supervised contrastive learning, and back-translation despite struggles with intervention data and deep semantic under-standing. Issues include truncated evidence from extended clinical trial premises (Kong et al., 2022)and insufficient model depth for causal reasoning. Future research should enhance semantic comprehension and causal reasoning and refine contrastive learning to improve the handling complex data and interventions, aiming to overcome current limitations in safe biomedical NLI.

5 Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant

Nos. 61966038 and 62266051. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902.

Nancy E Avis, Kevin W Smith, Carol L Link, Gabriel N Hortobagyi, and Edgardo Rivera. 2006. Factors associated with participation in breast cancer treatment clinical trials. *J Clin Oncol*, 24(12):1860–1867.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Eric Lehman, Benjamin Nye, Iain Marshall, and Byron C. Wallace. 2020. Evidence inference 2.0: More data, better models. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 123–132, Online. Association for Computational Linguistics.

Chao Feng, Jin Wang, and Xuejie Zhang. 2023. YNU-HPCC at SemEval-2023 task7: Multi-evidence natural language inference for clinical trial data based a BioBERT model. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 664–670, Toronto, Canada. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Jun Kong, Jin Wang, and Xuejie Zhang. 2022. Hierarchical bert with an adaptive fine-tuning strategy for document classification. *Knowledge-Based Systems*, 238:107872.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2023. Building efficient universal classifiers with natural language inference. *ArXiv*, abs/2312.17543.

xiaobo liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 10890–10905. Curran Associates, Inc.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Damien Sileo. 2023. tasksource: A dataset harmonization framework for streamlined nlp multi-task learning and evaluation. *ArXiv*, abs/2301.05948.

Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. 2020. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1):17.

Juraj Vladika and Florian Matthes. 2023. Sebis at SemEval-2023 task 7: A joint system for natural language inference and evidence retrieval from clinical trial reports. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1863–1870, Toronto, Canada. Association for Computational Linguistics.

Mengyuan Zhang, Jin Wang, and Xuejie Zhang. 2020. Using a pre-trained language model for medical named entity extraction in chinese clinic text. In *2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 312–317.

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. THiFLY research at SemEval-2023 task 7: A multi-granularity system for CTR-based textual entailment and evidence retrieval. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1681–1690, Toronto, Canada. Association for Computational Linguistics.

YNU-HPCC at SemEval-2024 Task 1: Self-Instruction Learning with Black-box Optimization for Semantic Textual Relatedness

Weijie Li, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
liweijie01@stu.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

This paper introduces a system designed for SemEval-2024 Task 1 that focuses on assessing Semantic Textual Relatedness (STR) between sentence pairs, including its multilingual version. STR, which evaluates the coherence of sentences, is distinct from Semantic Textual Similarity (STS). However, Large Language Models (LLMs) such as ERNIE-Bot-turbo, typically trained on STS data, often struggle to differentiate between the two concepts. To address this, we developed a self-instruction method that enhances their performance distinguishing STR, particularly in cases with high STS but low STR. Beginning with a task description, the system generates new task instructions refined through human feedback. It then iteratively enhances these instructions by comparing them to the original and evaluating the differences. Utilizing the Large Language Models' (LLMs) natural language comprehension abilities, the system aims to produce progressively optimized instructions based on the resulting scores. Through our optimized instructions, ERNIE-Bot-turbo exceeds the performance of conventional models in Track A,achieving a score enhancement of 4 to 7% on multilingual development datasets.

1 Introduction

SemEval-2024 Task 1(Ousidhoum et al., 2024) addresses the challenge of Semantic Textual Relatedness (STR), which goes beyond paraphrasing and entailment of Semantic Textual Similarity (STS)(Agirre et al., 2012, 2016; Cer et al., 2017; Xu et al., 2015) by considering topics and logical connections between sentence pairs. This task is particularly complex due to the nuanced context required for STR, a feature not fully captured by existing models trained predominantly on STS data. This gap can lead to black-box Large Language Models (LLMs) misinterpretations like ERNIE-Bot-turbo[1].

Our study introduces a self-instruction method to enhance the distinction between STS and STR in LLMs(Chen et al., 2023; Zhang et al., 2023; Hou et al., 2022; Wei et al., 2021). In our approach, back translation(Sennrich et al., 2016) converts low-resource language sentence pairs into English as inputs for LLMs. With a task description as the starting point, the black-box LLMs generate a new task instruction, which will be refined based on human feedback. The system iteratively refines the enhanced instruction by assessing it against the original and using the resulting score to produce increasingly optimized instructions. Our method improves how LLMs deal with tricky cases of similar but unrelated texts. Using our optimized instructions, ERNIE-Bot-turbo outperforms standard models and boosts scores by 4 to 7% on multilingual development datasets in Track A. The ranking of each Tack A's test dataset is as follows: English (36), Amharic (11), Algerian Arabic (24), Telugu (24), Spanish (24), Moroccan Arabic (24), Marathi (25), Kinyarwanda (20), and Hausa (20). The remainder of this paper is organized as follows. Section 2 describes the model and method used in our system, Section 3 discusses the results of the experiments, and finally, conclusions are drawn in Section 4.

2 Methodology

Figure 1 illustrates the overall framework of our self-instruction method. We employ back translation for datasets encompassing multiple languages to render sentence pairs into English as the input for LLMs. With a task description as the starting point, the black-box LLMs generates a new task instruction which will be refined based on human feedback. The enhanced instruction is subsequently assessed against the original, generat-

[1]https://yiyan.baidu.com/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 792–799

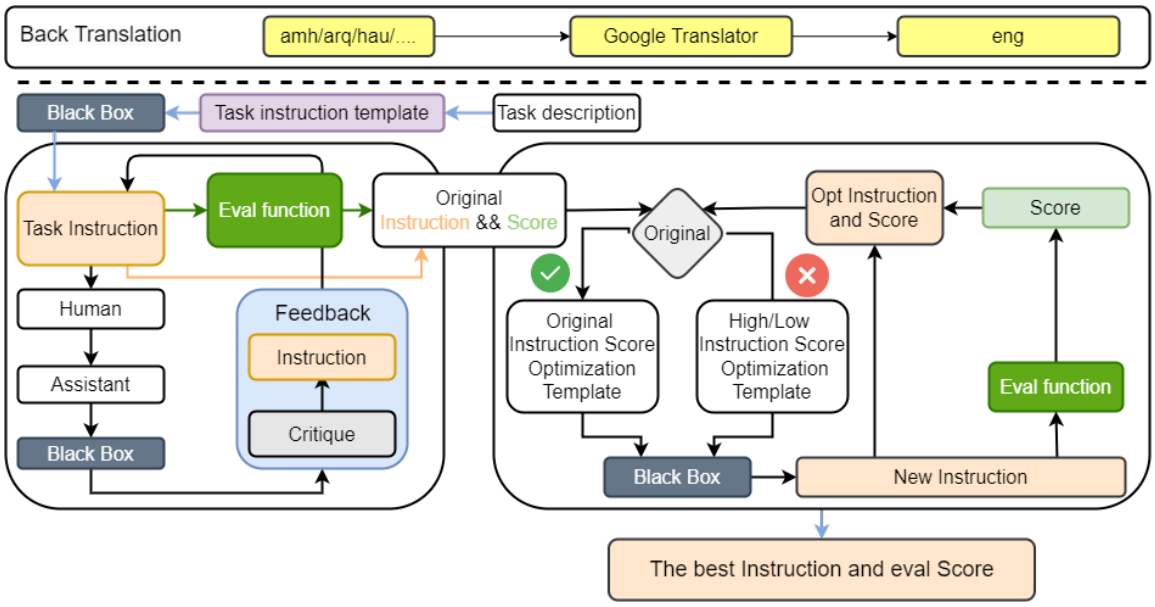


Figure 1: The framework of self-instruction method

ing a score that informs the optimization cycle. The system progressively refines the instructions in response to this score, resulting in progressively more optimized directives. The ensuing section will delve into a detailed analysis of this iterative optimization process.

2.1 Design of task instruction

Using sentences from the Amharic dataset as examples of hard sample with high Semantic Textual Similarity (STS) yet subtle Textual Relatedness (STR) for instance, What made him so certain? What contributed to his happiness? (original Amharic: "**ይህን ያህል እርግጠኛ እንዲሆን ያደረገው ምንድን ነው? ደስተኛ እንዲሆን አስተዋጽኦ ያደረገው ምንድን ነው?**"; goal label: 0.39) we underscore the significance of three components: instruction, the Chain of Thought (CoT)(Wei et al., 2022), and easily confused examples(Zhang et al., 2022; Li and Qiu, 2023). Human generated instructions aid LLMs in grasping the primary task but may not adequately explicate the concept of semantic textual relatedness (Figure 2.a) (Pred Score: 0.83). The CoT process facilitates LLMs in logical reasoning and analyzing sentence pairs, yet it encounters obstacles with complex samples prone to creating illusions (Figure 2.b) (Pred Score: 0.77). Easily confused examples are practical in dissecting hard samples but can skew the assessment of standard samples (Figure 2.c) (Pred Score: 0.67). Consequently, merging these approaches could provide more practical guidance for LLMs in discerning the relatedness of sentence pairs (Figure 2.d) (Prediction Score: 0.35). Detailed findings from the ablation study are discussed in Section 3.

2.2 Two fundamental components to generate the task instruction

Making use of natural language task description. LLMs excel in understanding natural language and simplifying the definition of optimization tasks. Capitalizing on this, we employ LLMs to convert the initial task description into detailed task instruction, guiding the LLMs to perform tasks such as STR analysis effectively, as indicated in Figure 3.a.

Refining task instruction through human feedback and evaluating their performance. While Large Language Models (LLMs) can generate task instructions from description, these instructions often fall short of being optimal and thus require human refinement and critical feedback. For instance, LLMs may overlook the significance of high and low relatedness (Figure 3.b). Subsequently, the improved instructions are evaluated, and their scores and instructions are integrated into the original framework, streamlining the subsequent optimization process (Figure 3.c).

2.3 LLMs as the black-box optimizer

After obtaining the original instruction-score (Figure 4.a), we utilize LLMs as the black-box optimizer to update and optimize the instructions iteratively. In each optimization step, the optimizer

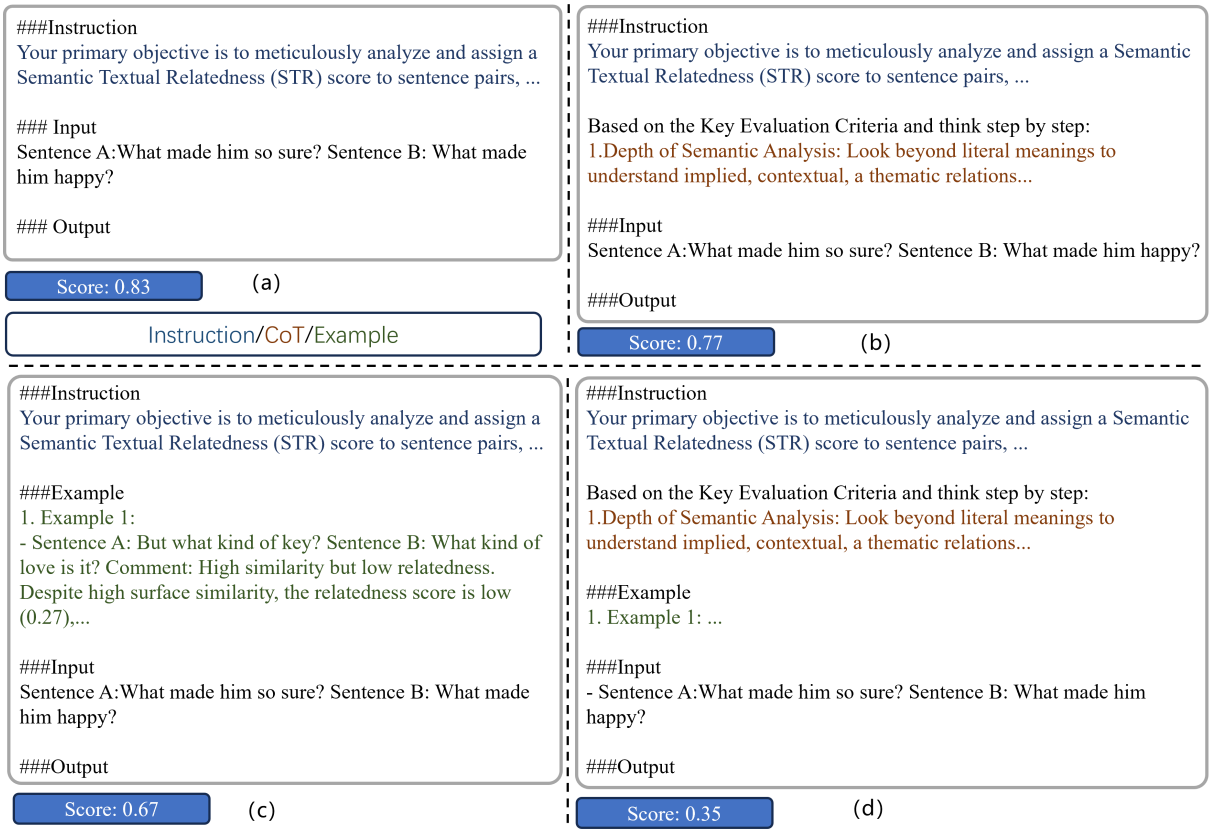


Figure 2: The process of task instruction design. (a)instruction(b)instruction + chain of thought(c)instruction + easily confused examples(d)instruction + chain of thought + easily confused examples

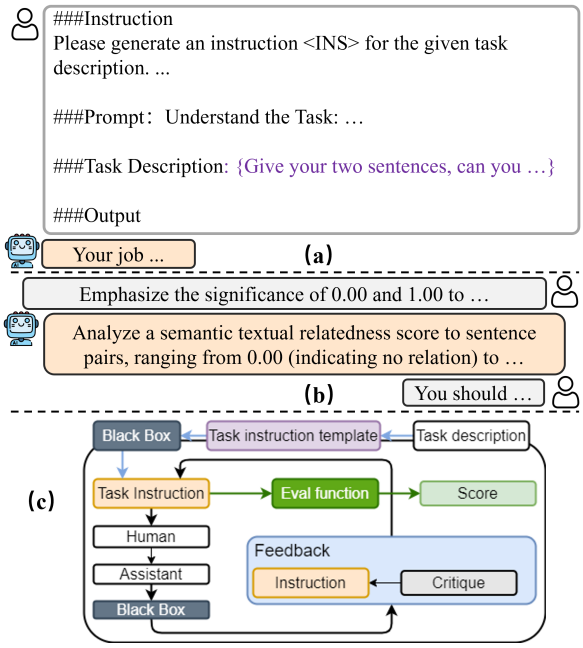


Figure 3: The process of original instruction optimization:(a)task instruction(b)task instruction optimization (c)overall process

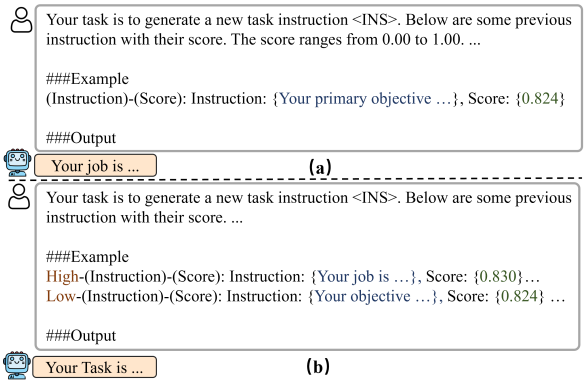


Figure 4: The process of utilizing LLM as the black-box optimizer

LLM generates candidate optimal instructions by analyzing the strengths of high-scoring and the weaknesses of low-scoring instructions (Figure 4.b). Subsequently, the new instruction is evaluated and added to the instruction-score list for sorting. From the instruction-score list, the top five high-scoring and the bottom five low-scoring instructions are selected and added to the instruction optimization template. The optimization process continues until the LLMs cannot propose new solutions with better optimization scores or the maximum number of optimization steps is reached.

Table 1: The evaluation scores of representative models from the four model methods on the training set.

BERT		Dual Sentence Encoding	
Model	Score	Model	Score
bert-base-uncased	0.673	all-mpnet-base-v2	0.787
bert-large-uncased	0.609	**all-MiniLM-L6-v2**	**0.824**
distilbert-base-uncased	0.673	all-MiniLM-L12-v2	0.816
deberta-base	0.668	all-distilroberta-v1	0.802
deberta-large	**0.678**	sentence-t5-base	0.805
deberta-large-mnli	0.659	sentence-t5-large	0.81
deberta-xlarge-mnli	0.651	sentence-t5-xl	0.805
distilroberta-base	0.618	moco-sentencebertV2.0	0.797
roberta-base	0.635		
roberta-large	0.44		
roberta-large-mnli	0.439		
Contrastive Learning		**LLM**	
Model	Score	Model	Score
sup-SimCSE-VietNamese-phobert-base	0.64	t5-base	0.705
sup-simcse-roberta-large	0.743	t5-large	0.702
sup-simcse-roberta-base	0.744	flan-t5-base	0.665
sup-simcse-bert-base-uncased	**0.8**	flan-t5-large	0.679
unsup-simcse-roberta-large	0.769	ERNIE-Bot-turbo(w/o opt)	0.782
diffcse-bert-base-uncased-sts	0.783	**ERNIE-Bot-turbo(w/ opt)**	**0.883**
diffcse-bert-base-uncased-trans	0.761		
diffcse-roberta-base-sts	0.774		
diffcse-roberta-base-trans	0.78		
esimcse-bert-base-uncased	0.778		
esimcse-bert-large-uncased	**0.798**		
esimcse-roberta-base	**0.792**		
esimcse-roberta-large	0.764		
pcl-bert-base-uncased	0.776		
pcl-bert-large-uncased	**0.799**		
pcl-roberta-base	0.766		
pcl-roberta-large	0.755		

3 Experimental Result

Datasets. The STR task dataset comprises datasets in 14 distinct languages, including 9 languages specifically for Track A. Each language dataset contains pairs of sentences, where each pair in the training, development, and test sets is assigned a gold score. This score reflects the degree of STR between the two sentences, ranging from 0 to 1, as determined by manual annotation. Figure 5 below presents the composition of the training, test, and development sets for Track A.

Evaluation Metrics. The STR in Track A is evaluated using the spearman rank correlation coefficient(Sedgwick, 2014), which measures how well the system predicted rankings of test instances align with human judgment. The metric will be calculated as follows:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \quad (1)$$

where d_i represents the difference between the ranks of the i-th pair of sentences,n is the number of pairs of sentences,ρ is the spearman rank correlation.

3.1 Implementation Details

Our approach, addressing the scarcity of low-resource languages, uses back translation to convert their sentence pairs into English for (LLMs) inputs. This experiment prioritizes scoring on English datasets to select the most effective score model. We assess four baseline methods: BERT(Devlin et al., 2019; Sanh et al., 2019; He et al., 2020; Delobelle et al., 2020; Raffel et al., 2020; Chung et al., 2022), dual sentence encoding(Reimers and Gurevych, 2019; Ni et al., 2022), contrastive learning(Gao et al., 2021; Song et al., 2020; Wang et al., 2020; Chuang et al., 2022; Wu et al., 2022b,a) , and LLMs. These models were evaluated using the training set, with results presented in Table 1. Considering our experimental objective of analyzing hard samples and scoring sentence pair STR, ERNIE-Bot-turbo was chosen as the scoring model. The LLMs utilized as the optimizer and scorer are: (a)optimizer LLM: gpt-3.5-turbo and (b)scorer LLM: ERNIE-Bot-turbo.

3.2 Desigin of task instruction

At the experiment's outset, we performed adaptation tests on the English training dataset using four variations of instruction templates: (1) instruction only, (2) instruction with chain-of-thought, (3) instruction with easily confused examples, and (4) instruction with both chain-of-thought and easily confused examples. The experimental results in Figure 6 suggest that combining instruction, chain-of-thought, and easily confused examples significantly aids LLMs in semantic textual relatedness analysis.

3.3 Prompt optimization

The score LLM operates at a temperature of 0, ensuring deterministic decoding, whereas the optimizing LLM uses a temperature of 0.95 promoting creativity in instruction generation. Figure 7.a illustrates the accuracy fluctuations during the model's evaluation on the English training dataset. Figure 7.b presents the scores for Track A's development in three scenarios: without optimization, optimized (val-score: 0.8360) and further optimized (val-score: 0.8839). Figure 8 delves into the impact of these three optimization scenarios on hard samples. Consequently, our methodology effectively reduces the hallucinations of LLMs in STS and STR tasks. This leads to a more comprehensive analysis of hard samples and consistently

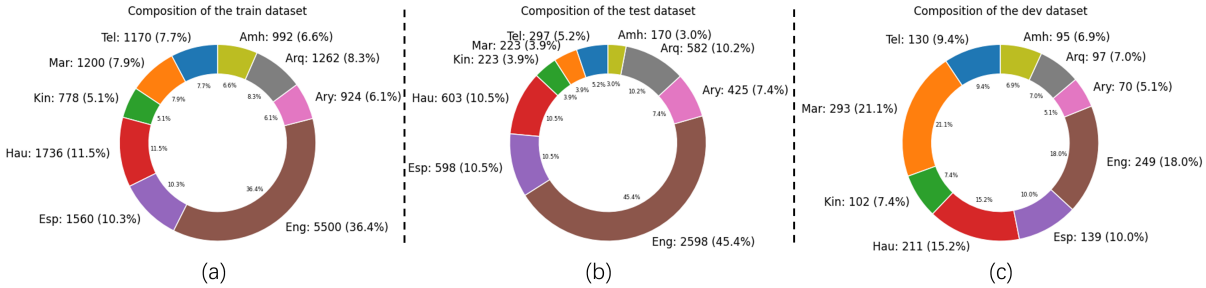


Figure 5: The composition of the Track A's training(a), test(b), and development(c) dataset

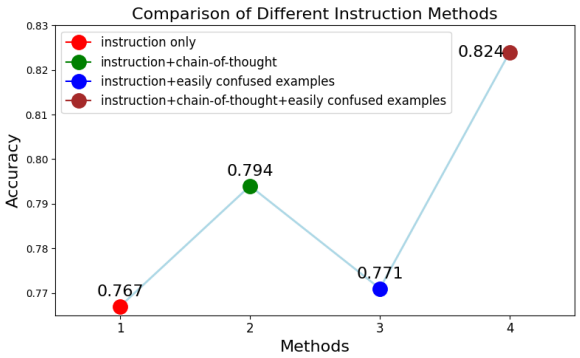


Figure 6: The ablation tests for four variations of instruction templates: 1.instruction 2. instruction + chain of thought 3.instruction + easily confused examples 4.instruction + chain of thought + easily confused examples

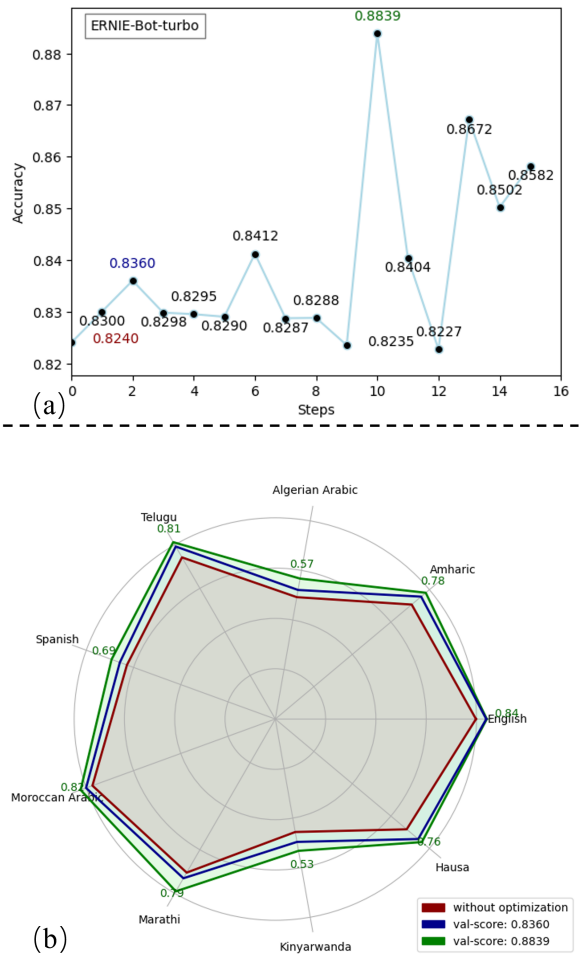


Figure 7: (a) shows changes in accuracy during evaluation on the English training dataset. (b) shows the development scores for Track A in scenarios: without optimization, optimized (val-score: 0.8360), and further optimized (val-score: 0.8839).

improves performance evaluations on the training dataset through an iterative process.

3.4 Result and Discussion

Results. Our final evaluation compared the 'no optimization' approach to 'optimization' across Track A's nine language development datasets using back translation, as shown in Figure 9.The outcomes indicate that optimized instructions significantly enhanced performance by 4 to 7% over the non-optimized approach.The ranking of each test dataset are as follows: English (36), Amharic (11), Algerian Arabic (24), Telugu (24), Spanish (24), Moroccan Arabic (24), Marathi (25), Kinyarwanda (20), and Hausa (20).

Discussion. The experimental results suggest the following:

- Our self-instruction method effectively reduces confusion between STS and STR in Large Language Models (LLMs), thereby improving accuracy and enhancing the LLMs's capability to analyze standard samples, particularly in examining hard sample.
- However, the experimental outcomes are somewhat modest due to the coarse granularity of the STR task and the considerable overlap between semantic textual similarity and relatedness.

##Instruction
Your task is to predict a relatedness score between two sentences based on their semantic relatedness. …

Based on the Key Evaluation Criteria and think step by step:
1. Depth of Semantic Analysis: ...

###Examples and Analysis:
1. Example 1:SentenceA:But what kind of key? SentenceB:What kind of love is it?
Comment:Highly similar but low relatedness (Relatedness score: 0.27).
2. Example 2: ...

###Input
SentenceA: What made him so sure? Sentence B： What made him happy?

###Output

Score: 0.67

(without any optimization)

##Instruction
Our responsibility is to meticulously analyze and assign a Semantic Textual Relatedness (STR) score between sentence pairs, …

Based on the Key Evaluation Criteria and think step by step
1.Depth of Semantic Analysis: ...
2.Significance of Relatedness: ...

###Examples and Analysis
1. Example 1:
-SentenceA:But what kind of key? SentenceB:What kind of love is it?
Comment:(Highly similarity,but low relatedness.the relatedness score is 0.27)
2. Example 2: ...

###Input
SentenceA: What made him so sure? SentenceB: What made him happy?

###Output

Score: 0.55 **(val-score : 0.8360)**

##Instruction
Your primary objective is to meticulously analyze and assign a Semantic Textual Relatedness (STR) score to sentence pairs, ranging from 0.00 ...
Emphasize the significance of differentiating ...

Based on the Key Evaluation Criteria and think step by step:
1. Depth of Semantic Analysis: ...
2. Significance of Relatedness: ...
1.Low Relatedness: ...

###Examples and Analysis:
1. Example 1:
SentenceA:But what kind of key? SentenceB:What kind of love is it?Comment: Despite high surface similarity, the relatedness score is low (0.27), indicating a lack of substantial semantic connection...
...

###Input
SentenceA: What made him so sure? Sentence B： What made him happy?

###Output

(val-score: 0.8839)

Score: 0.42

Figure 8: It demonstrates how the scoring model assesses the impact of these three optimization scenarios on hard samples.(red: score, brown: chain-of-thought optimization,blue: example analysis optimization, purple: instruction optimization)

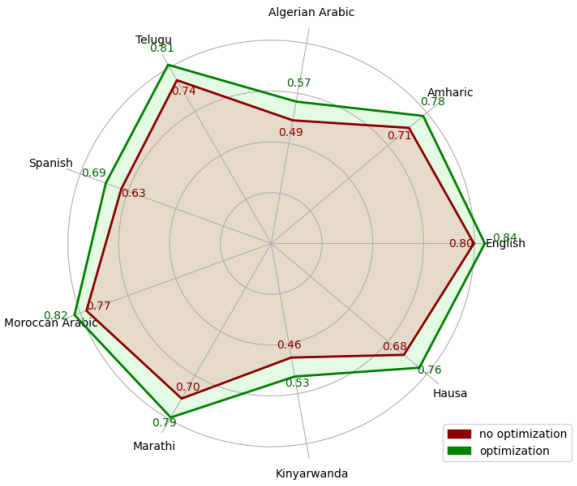


Figure 9: The performance of no optimization and optimization in development dataset.

- The back translation method encounters notable challenges when utilized with low-resource languages such as Arabic. This is primarily due to significant language biases between low-resource and high-resource languages like English within the semantic space, directly influencing the scoring model's judgment.

- The limitation of the score model is still an obstacle to performance. ERNIE-Bot-turbo (score model), trained on Chinese and English datasets corpus, demonstrates weaker proficiency in evaluating English sentence pairs.

4 Conclusion

In this paper, we developed a self-instruction method that enhances LLMs' ability to distinguish between Semantic Textual Similarity (STS) and Semantic Textual Relatedness (STR), particularly in hard samples(High STS but low STR).Through this method, ERNIE-Bot-turbo (score LLM) not only surpasses the performance of conventional models, achieving a score enhancement of 4 to 7 % on multilingual development datasets, but also effectively reduces confusion between STS and STR in Large Language Models (LLMs). Additionally, it achieved a commendable ranking in the final test evaluation. Our work demonstrates that optimized instructions, chain of thought, and easily confused examples enable LLMs to mitigate errors even in few-shot samples. Future research will aim to refine LLMs' capacity to grasp the overall semantic meaning of sentences further.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments.This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos.61966038 and 62266051.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2023. Instructzero: Efficient instruction optimization for black-box large language models. *arXiv preprint arXiv:2306.03082*.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Yutai Hou, Hongyuan Dong, Xinghao Wang, Bohan Li, and Wanxiang Che. 2022. MetaPrompting: Learning to learn better prompts. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3251–3262, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, et al. 2024. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *arXiv preprint arXiv:2402.08638*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Philip Sedgwick. 2014. Spearmans rank correlation coefficient. *Bmj*, 349.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022a. Pcl: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. *arXiv preprint arXiv:2201.12093*.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022b. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhihan Zhang, Shuohang Wang, Wenhao Yu, Yichong Xu, Dan Iter, Qingkai Zeng, Yang Liu, Chenguang Zhu, and Meng Jiang. 2023. Auto-instruct: Automatic instruction generation and ranking for black-box language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9850–9867, Singapore. Association for Computational Linguistics.

AAdaM at SemEval-2024 Task 1: Augmentation and Adaptation for Multilingual Semantic Textual Relatedness

Miaoran Zhang[1] **Mingyang Wang**[2,3] **Jesujoba O. Alabi**[1] **Dietrich Klakow**[1]
[1]Saarland University, Saarland Informatic Campus
[2]Bosch Center for AI, [3]LMU Munich
mzhang@lsv.uni-saarland.de

Abstract

This paper presents our system developed for the SemEval-2024 Task 1: Semantic Textual Relatedness for African and Asian Languages. The shared task aims at measuring the semantic textual relatedness between pairs of sentences, with a focus on a range of underrepresented languages. In this work, we propose using machine translation for data augmentation to address the low-resource challenge of limited training data. Moreover, we apply task-adaptive pre-training on unlabeled task data to bridge the gap between pre-training and task adaptation. For model training, we investigate both full fine-tuning and adapter-based tuning, and adopt the adapter framework for effective zero-shot cross-lingual transfer. We achieve competitive results in the shared task: our system performs the best among all ranked teams in both subtask A (supervised learning) and subtask C (cross-lingual transfer).[1]

1 Introduction

Semantic Textual Relatedness (STR) measures the closeness of meaning between two linguistic units, such as a pair of words or sentences (Budanitsky, 1999; Mohammad and Hirst, 2012). For example, one can easily tell that *"I like playing games"* is more semantically related to *"The game is fun"* rather than *"The weather is good"*, which largely depends on their lexical semantic relation and topic consistency. Semantic Textual Similarity (STS), a closely related concept, indicates whether two units have a paraphrasing relation. The difference between these two concepts is clarified in Abdalla et al. (2023): while similar pairs are also related, the reverse is not necessarily true.

In stark contrast to the extensive research on STS (Gao et al., 2021; Chuang et al., 2022; Zhang et al., 2022; Seonwoo et al., 2023), exploration of STR lags behind and predominantly focuses on English (Marelli et al., 2014; Abdalla et al., 2023), mainly due to the lack of datasets. To close this gap, the SemEval-2024 Task 1: Semantic Textual Relatedness (Ousidhoum et al., 2024b) is proposed to encourage STR research on 14 African and Asian languages. The shared task consists of 3 subtasks: supervised (subtask A), unsupervised (subtask B), and cross-lingual (subtask C).

In this paper, we present our system AAdaM (Augmentation and Adaptation for Multilingual STR) developed for subtask A and C. Our system adopts a cross-encoder architecture which takes the concatenation of a pair of sentences as input and predicts the relatedness score through a regression head (Devlin et al., 2019). As the provided task data for non-English languages is relatively limited, we perform data augmentation for these languages via machine translation. To better adapt a pre-trained model to the STR task, we apply task-adaptive pre-training (Gururangan et al., 2020) which has shown effectiveness on many tasks (Xue et al., 2021; Wang et al., 2023). For subtask A, we explore full fine-tuning and adapter-based tuning (Houlsby et al., 2019) combined with previously mentioned techniques. Additionally, we use the adapter framework MAD-X (Pfeiffer et al., 2020) for cross-lingual transfer in subtask C.

We select the best model based on the performance on development sets for the final submission, and our system achieves competitive results on both subtasks. In subtask A, our system ranks first out of 40 teams on average, and performs the best in Spanish. In subtask C, our system ranks first among 18 teams on average, and achieves the best performance in Indonesian and Punjabi.

2 SemRel Dataset

To encourage STR research in the multilingual context, Ousidhoum et al. (2024a) introduce SemRel, a new STR dataset annotated by native speakers, covering 14 languages from 5 distinct lan-

[1]Our code: https://github.com/uds-lsv/AAdaM

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 800–810

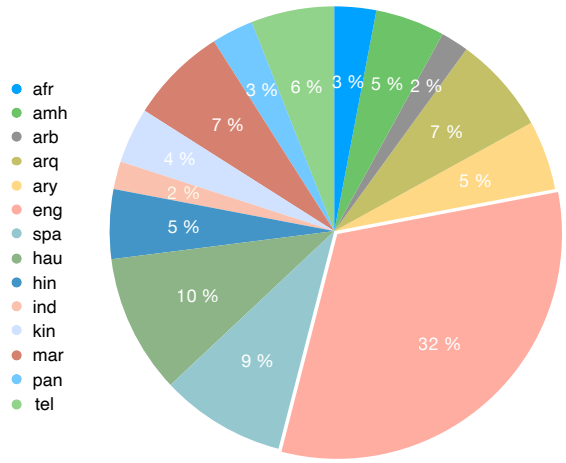


Figure 1: SemRel data distribution across languages.

guage families. These languages are mostly spoken in Africa and Asia, and many of them are under-represented in natural language processing resources. As shown in Figure 1, the data sizes vary widely from language to language constrained by the availability of resources. Notably, English data comprises 32% of the whole dataset and surpasses other languages by a large margin.

3 System Overview

Our system employs a *cross-encoder* architecture, which takes the concatenation of a pair of sentences as input and predicts the relatedness score through a regression head. Compared to bi-encoders (Reimers and Gurevych, 2019), which extract individual sentence representations and then compare them using cosine similarity, cross-encoders generally perform better, at the cost of increased inference latency (Humeau et al., 2020). We select cross-encoder because of its superior performance (see Appendix A), and leave the exploration of an efficient alternative as future work.

The core techniques underlying our system are (i) **data augmentation** using machine translation (§3.1), and (ii) **task-adaptive pre-training** on unlabeled task data (§3.2). We explore two training paradigms for supervised learning combined with the aforementioned techniques, i.e., **fine-tuning** and **adapter-based tuning** (§3.3), and the latter is also employed for cross-lingual transfer (§3.4).

3.1 Data Augmentation

Data augmentation (DA) serves as a widely used strategy to mitigate data scarcity in low-resource languages (Hedderich et al., 2021; Feng et al., 2021). Inspired by work on DA with machine translation (Hu et al., 2020; Amjad et al., 2020), we create additional training data for non-English languages by translating from various English sources, as illustrated below.

SemRel translation. As English data occupies a significant portion of the entire SemRel dataset, we perform augmentation by translating the English subset to other target languages.

STS-B translation. STS-B (Cer et al., 2017), a semantic similarity dataset, is highly relevant to STR, and we translate the STS-B training set in English to other target languages.

It worth noting that using translations as data augmentation yields a mixed data quality. For instance, the translation process may introduce artifacts that reduce data validity. Additionally, the concepts of "similarity" and "relatedness" are relevant but not equivalent, leading to a mismatch in their annotated scores. To leverage data in varied qualities, Zhu et al. (2023) shows that a two-phase approach is beneficial, in which the model is trained on noisy data first and then trained on clean data. Our training procedure follows this two-phase scheme: (i) training the model on augmented data as a *warmup*, and (ii) subsequently training the model on the original task data.

3.2 Task-Adaptive Pre-training

Pre-trained language models (PLMs) are trained on massive text corpora with self-supervision objectives for general purposes (Devlin et al., 2019; Liu et al., 2019). To better adapt PLMs to downstream tasks, Gururangan et al. (2020) propose task-adaptive pre-training (TAPT), i.e., continued pre-training on task-specific unlabeled data, and show that it can effectively improve downstream task performance. We integrate this strategy into our system, wherein we conduct masked language modeling (MLM) on unlabeled task data for a given target language before initiating any supervised training.

3.3 Fine-tuning vs. Adapter-based Tuning

Fine-tuning is the conventional approach to adapt general-purpose PLMs to downstream tasks. It updates all model parameters for each task, leading to inefficiency with the ever-increasing model scales and number of tasks. Recently, many works focus on introducing lightweight alternatives to improve parameter efficiency (Lester et al., 2021; Hu et al., 2022; He et al., 2022). For example, adapter-based

Model Tuning	TAPT	Warmup	arq	amh	eng	hau	kin	mar	ary	spa	tel
FINE-TUNING	✗	✗	52.96	87.70	83.07	78.91	68.59	85.23	88.26	73.83	84.90
	✗	SemRel	55.96	87.86	/	79.87	70.06	85.51	**88.59**	72.93	85.38
	✗	STS-B	62.05	88.50	**84.31**	79.86	69.78	86.48	86.97	73.33	85.15
	✓	✗	65.70	88.03	82.79	79.41	67.03	84.88	88.50	70.47	83.84
	✓	SemRel	66.74	85.58	/	**80.73**	71.29	85.74	87.01	73.37	**85.77**
	✓	STS-B	**68.25**	**88.72**	83.01	78.95	69.38	85.26	87.07	73.50	84.66
ADAPTER TUNING	✗	✗	55.44	87.01	82.96	78.23	70.45	84.62	86.43	72.62	84.51
	✗	SemRel	59.58	87.66	/	79.15	70.56	86.54	86.88	74.90	84.88
	✗	STS-B	62.83	87.63	82.97	80.29	82.01	87.18	87.53	74.18	84.17
	✓	✗	58.81	85.61	82.74	78.40	70.48	84.56	85.78	72.15	84.34
	✓	SemRel	58.47	87.57	/	79.78	71.67	**87.24**	87.35	**76.65**	85.69
	✓	STS-B	59.58	87.40	82.32	79.22	**73.04**	87.12	87.22	73.22	83.70

Table 1: Subtask A performance on development sets (Spearman's correlation $\times 100$). SemRel: warmup by training on SemRel translations; STS-B: warmup by training on STS-B translations. We underline the best performance of fine-tuning and adapter-based tuning, and **bold** the best performance across all variants.

tuning (Houlsby et al., 2019) only updates small modules known as adapters inserted between the layers of PLMs while keeping the remaining parameters frozen. In particular, it has shown impressive performance in cross-lingual transfer (Pfeiffer et al., 2020; Ansell et al., 2021; Pfeiffer et al., 2022).

We explore both fine-tuning and adapter-based tuning to compare their effectiveness on multilingual STR. For fine-tuning, we update all model parameters at each stage, namely the TAPT stage, the warmup stage and the final training stage using the original task data. For adapter-based tuning, we utilize the MAD-X framework (Pfeiffer et al., 2020) which consists of language-specific adapters and task-specific adapters. The language adapters are pre-trained with an MLM objective on unlabeled monolingual corpora. To this end, we collect open-source data from the Leipzig Corpus Collection (Goldhahn et al., 2012) for pre-training.[2] The task adapters are trained on labeled task-specific data (augmented or original), while keeping the language adapters fixed. Note that when applying TAPT, only language adapters are updated. In subtask A, we apply fine-tuning and adapter-based tuning in combination with TAPT and warmup techniques, and select the best model based on the performance on development sets.

3.4 Cross-lingual Transfer with Adapters

The high modularity of MAD-X enables efficient zero-shot cross-lingual transfer. During inference, we simply replace the source language adapter with the *target language adapter* while retaining the *source task adapter*. This task adapter has been trained on labeled data from the source language, without prior exposure to the target language.[3] A crucial challenge for cross-lingual transfer lies in source language selection, as improper sources may lead to negative results (Lange et al., 2021). To determine the best source language, we explore the following metrics to rank sources: (1) linguistic distance (Littell et al., 2017), (2) token overlap (Wu and Dredze, 2019), and (3) development set performance.[4] Results in Appendix C demonstrate that development set performance serves as the most reliable indicator of transfer performance. For subtask C, we select the optimal source from the adapters trained in subtask A based on their performance on development sets.

4 Experimental Setup

Model. Our backbone model is AfroXLMR-large-61L (Adelani et al., 2024), adapted from XLM-R (Conneau et al., 2020) through multilingual adaptive fine-tuning (Alabi et al., 2022). We use NLLB (`nllb-200-distilled-600M`) (Team et al., 2022) to translate from English resources to other languages as data augmentation.

Implementation. All experiments are conducted on a single NVIDIA A100 GPU with a batch size of 16. For MLM, we set the learning rate to 5e-5

[2]Details are provided in Appendix B.

[3]Note that when transferring from any other language to English, we ensure that the source task adapter has not been trained on augmented data translated from English resources, thereby eliminating the effect of data leakage.

[4]The existence of development sets is not realistic in the true zero-shot scenario, and we leave further discussion to the Limitations section.

Model	arq	amh	eng	hau	kin	mar	ary	spa	tel	Avg.↑
Overlap♢	40.	63.	67.	31.	33.	62.	63.	67.	70.	55.11
LaBSE♢	60.	85.	83.	69.	72.	88.	77.	70.	82.	76.22
PALI	67.88	**88.86**	**86.00**	**76.43**	81.34	**91.08**	**86.26**	72.38	86.43	**81.85**
king001	**68.23**	88.78	84.30	74.72	**81.69**	89.68	85.97	72.12	85.34	81.20
NRK	67.36	86.42	83.29	67.20	75.69	87.93	82.70	68.99	83.42	78.11
saturn	57.77	84.51	-	69.91	75.53	87.28	79.77	-	**87.34**	-
AAdaM (Ours)	66.23	86.71	84.84	72.36	77.91	89.43	83.50	**74.04**	84.77	79.98

Table 2: Subtask A performance on test sets (Spearman's correlation ×100). ♢: baseline results from Ousidhoum et al. (2024a). We **bold** the best performance across submitted systems.

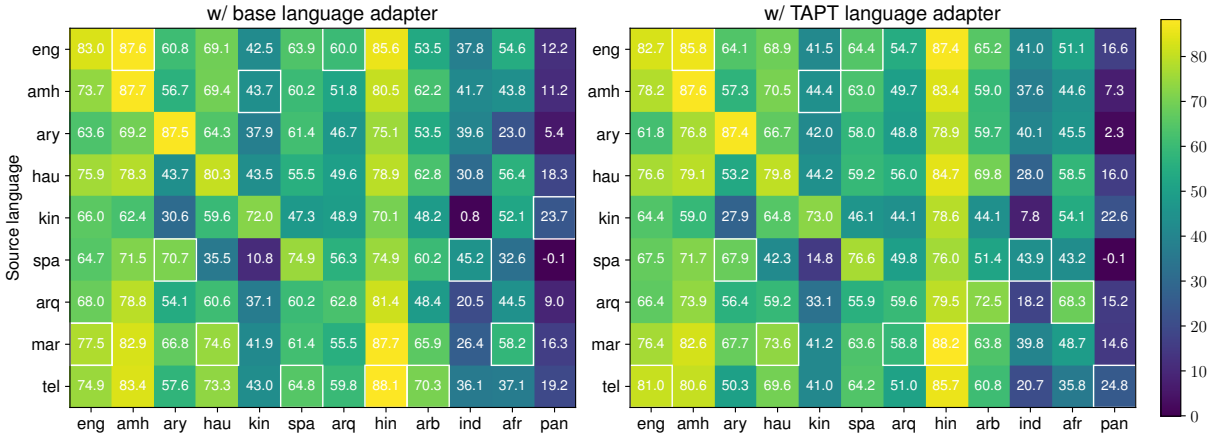


Figure 2: Subtask C performance on development sets (Spearman's correlation ×100) using different types of language adapters. Boxes highlight the optimal performances for each target language, and we select the best source for final submission.

and train models for 10 epochs. For fine-tuning, we conduct a grid-search of learning rate from {2e-5, 5e-5} on SemRel development sets and train models for 6 epochs. For adapter-based tuning, we select the optimal learning rate from {1e-4, 2e-4, 5e-5} and train adapters for 15 epochs.

5 Results and Analysis

5.1 Subtask A: Supervised Learning

In Table 1, we compare the performance on development sets using fine-tuning and adapter-based tuning along with various techniques. Fine-tuning achieves the best performance in most languages (6 out of 9), which is unsurprising as it optimizes the entire parameter space. Notably, adapter-based tuning demonstrates comparable performance to fine-tuning in Hausa (hau) and Telugu (tel), while even surpassing it in Kinyarwanda (kin), Marathi (mar) and Spanish (spa). Looking at the effectiveness of TAPT and warmup, we observe that they provide benefits in most cases compared to using no techniques at all. Nonetheless, the improvements are sometimes marginal, particularly in languages such as Amharic (amh), English (eng), and Moroccan Arabic (ary), where the baseline performances are already relatively strong compared to other languages.

In our final submission, we selected the best model for each language based on the performance of development sets. As shown in Table 2, our approach largely improves the baseline results (Ousidhoum et al., 2024a), especially for Algerian Arabic (arq), Kinyarwanda (kin), and Moroccan Arabic (ary). In comparison to several top-performing submitted systems, we achieve the best performance in Spanish (spa). There were a total of 40 final submissions in subtask A, and our system ranks first on average in the official leaderboard.[5]

5.2 Subtask C: Cross-lingual Transfer

In subtask C, we replace source language adapters from subtask A with target language adapters. We analyze two groups of language adapters: base language adapters trained only on Leipzig corpora and TAPT language adapters further trained on unlabeled task data. The cross-lingual transfer results

[5]PALI and king001 also achieved competitive performance; however, they are not ranked in the official leaderboard due to missing system descriptions.

Model	afr	arq	amh	eng	hau	hin	ind	kin	arb	ary	pan	spa	Avg.↑
Overlap♢	71.	40.	63.	67.	31.	53.	55.	33.	32.	63.	-27.	67.	45.67
LaBSE♢	79.	46.	84.	80.	62.	76.	47.	57.	61.	40.	-5.	62.	57.42
king001	81.00	**61.44**	**87.83**	-	**73.35**	**84.39**	37.58	62.99	65.68	**81.96**	-	**70.76**	-
UAlberta	80.57	44.13	81.60	-	67.85	82.78	44.90	63.58	**67.15**	60.22	-1.74	57.16	-
ustcctsu	74.87	41.44	70.90	78.40	47.63	65.80	46.02	45.41	46.87	61.32	-24.79	68.51	51.87
umbclu	**82.23**	12.63	4.30	78.75	45.69	15.52	51.53	48.36	3.54	-3.75	-7.75	60.89	32.66
AAdaM (ours)	81.39	55.07	86.29	**79.37**	72.88	83.86	**52.80**	**64.99**	65.32	60.03	**15.53**	62.05	**64.97**

Table 3: Subtask C performance on test sets (Spearman's correlation ×100). ♢: baseline results from Ousidhoum et al. (2024a). We **bold** the best performance across submitted systems.

on development sets are shown in Figure 2. We observe a discrepancy in the optimal source languages selected with two types of adapters, indicating a behavior shift after applying TAPT. Furthermore, the performance for target languages shows high sensitivity to the choice of source language. For example, using Spanish (spa) as the source language for Indonesian (ind) performs significantly better than using Kinyarwanda (kin), showcasing the importance of careful source language selection. When examining each target language, we find that in the case of Amharic (amh), the cross-lingual transfer performance is comparable to its supervised learning performance. However, it remains a challenge for a few languages, such as Indonesian (ind) and Punjabi (pan).

The results for test sets are shown in Table 3. Compared to LaBSE (Feng et al., 2022), a multilingual sentence embedding model, our cross-lingual transfer approach achieves better performance on most languages, especially for Algerian Arabic (arq), Hausa (hau), Moroccan Arabic (ary), and Punjabi (pan). However, our system is surpassed by the simple word overlap baseline in Indonesian (ind), Moroccan Arabic (ary) and Spanish (spa). This highlights the need for nuanced investigation of data distributions across various languages. Subtask C received 18 submissions in total, and we perform the best in the official leaderboard. In particular, we achieve the best performance in Indonesian (ind) and Punjabi (pan), which seem harder for other teams. For Punjabi (pan), where most teams get negative correlation scores, our method maintains its effectiveness.

5.3 Analysis

We partition ground-truth relatedness scores, ranging from 0 to 1, to different levels for fine-grained analysis. Figure 3 shows the detailed model performance for several under-performing languages.

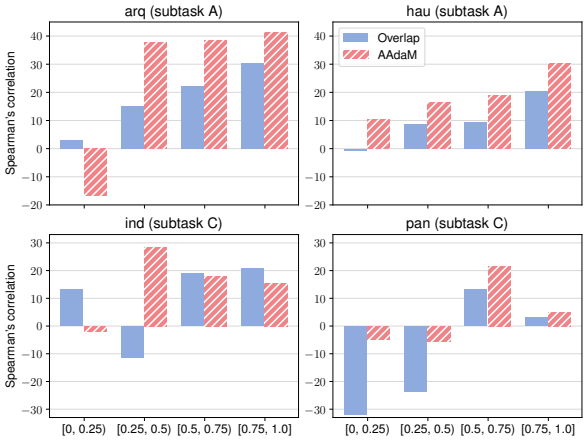


Figure 3: Performance on test sets (Spearman's correlation ×100) in different relatedness levels.

Although our evaluation scores on the entire test sets are all positive, some subsets exhibit negative correlations, particularly those with lower relatedness scores. Moreover, AAdaM largely lags behind the simple word overlap baseline for Algerian Arabic (arq) and Indonesian (ind) within the 0 to 0.25 range. These observations highlight the complexity of capturing nuanced relationships within specific categories, possibly affected by the data annotation procedure and unbalanced learning.

6 Conclusion

In this paper, we introduce our multilingual STR system, AAdaM, developed for the SemEval-2024 Task 1, which achieves competitive results in both subtask A and subtask C. We see noticeable improvements by using data augmentation and task-adaptive pre-training, and demonstrate that adapter-based tuning is an effective approach for supervised learning and cross-lingual transfer. Despite these strengths, our fine-grained analysis reveals that capturing nuanced semantic relationships remains a challenge, highlighting the need for further granular investigation and modeling improvements.

Limitations

Although our approach has demonstrated impressive performance, relying on development sets for source language selection undermines its practical value in the true zero-shot setting. While linguistic (dis)similarity (Littell et al., 2017) is a commonly used estimator for cross-lingual transfer performance, it alone does not explain many transfer results (Lauscher et al., 2020). Philippy et al. (2023) survey different factors that impact cross-lingual transfer performance, finding contradictory conclusions from previous studies. In future work, we plan to scrutinize the interplay among various factors, and select the optimal source language without relying on post-hoc evaluation.

Acknowledgements

We thank Vagrant Gautam and Badr M. Abdullah for their proofreading and anonymous reviewers for their feedback. Miaoran Zhang received funding from the DFG (German Research Foundation) under project 232722074, SFB 1102. Jesujoba O. Alabi was supported by the BMBF's (German Federal Ministry of Education and Research) SLIK project under the grant 01IS22015C.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

David Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba Alabi, Yanke Mao, Haonan Gao, and En-Shiun Lee. 2024. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta. Association for Computational Linguistics.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Maaz Amjad, Grigori Sidorov, and Alisa Zhila. 2020. Data augmentation using machine translation for fake news detection in the Urdu language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2537–2542, Marseille, France. European Language Resources Association.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexander Budanitsky. 1999. Lexical semantic relatedness and its application in natural language processing. Technical report, technical report CSRG-390, Department of Computer Science, University of Toronto.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Lukas Lange, Jannik Strötgen, Heike Adel, and Dietrich Klakow. 2021. To share or not to share: Predicting sets of sources for model transfer learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8744–8753, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International*

Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Saif M Mohammad and Graeme Hirst. 2012. Distributional measures as proxies for semantic relatedness.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Fred Philippy, Siwen Guo, and Shohreh Haddadan. 2023. Towards a common understanding of contributing factors for cross-lingual transfer in multilingual language models: A review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5877–5891, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Yeon Seonwoo, Guoyin Wang, Changmin Seo, Sajal Choudhary, Jiwei Li, Xiang Li, Puyang Xu, Sunghyun Park, and Alice Oh. 2023. Ranking-enhanced unsupervised sentence representation learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15783–15798, Toronto, Canada. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schütze. 2023. NLNDE at SemEval-2023 task 12: Adaptive pretraining and source language selection for low-resource multilingual sentiment analysis. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 488–497, Toronto, Canada. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Miaoran Zhang, Marius Mosbach, David Adelani, Michael Hedderich, and Dietrich Klakow. 2022. MCSE: Multimodal contrastive learning of sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5959–5969, Seattle, United States. Association for Computational Linguistics.

Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. 2023. Weaker than you think: A critical look at weakly supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14229–14253, Toronto, Canada. Association for Computational Linguistics.

A Model and Architecture Selection

In our preliminary study, we examine the capacity of different pre-trained models with or without any training. To assess their out-of-the-box effectiveness, we extract contextual embeddings for pairs of sentences from various multilingual models, and use the cosine similarity to predict the semantic relatedness score. The multilingual models include:

- **sentence transformers**: mpnet-base-v2[6] and LaBSE (Feng et al., 2022)
- **general-purpose models**: XLMR-large (Conneau et al., 2020), AfroXLMR-large (Alabi et al., 2022), AfriBERTa-large (Ogueji et al., 2021), AfroXLMR-large-61L and AfroXLMR-large-75L (Adelani et al., 2024)

Additionally, we add two simple baselines for comparison: word overlap[7] and fastText (Mikolov et al., 2018). For both fastText vectors and contextual embeddings, we employ mean pooling to get sentence embeddings.

In Table 4, we can see that sentence transformers achieve superior performance in most languages when no training is conducted. This observation is not unsurprising, as they have been trained for sentence embeddings that can better capture the semantic relationships. However, this trend shifts upon fine-tuning the models on task data with either bi-encoder or cross-encoder architecture. Notably, with the cross-encoder architecture, AfroXLMR-large-61L achieves comparable performance to LaBSE. To satisfy the requirement in subtask C, for which the pre-trained model should not be trained on any relatedness or similarity datasets, we adopt AfroXLMR-large-61L as our backbone model with the cross-encoder architecture for all our experiments.

[6] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

[7] https://github.com/semantic-textual-relatedness/Semantic_Relatedness_SemEval2024/blob/main/STR_Baseline.ipynb

B Pre-training Data Collection

To pre-train language adapters, we collect open-source corpora from the Leipzig Corpus Collection and use the recent data derived from news and wikipedia domains. Data statistics are shown Table 5. As the SemRel data spans over diverse domains, there is a potential risk of domain mismatch between the pre-training data and task data, which needs a further investigation.

C Source Language Selection

To determine the best source language for cross-lingual transfer, we explore three metrics to estimate the transfer performance:

Linguistic distance. We use the average of six distances obtained from the URIEL Database (Littell et al., 2017) to measure the similarity between a pair of languages. These distances include syntactic, phonological, inventory, geographic, genetic, and featural distances. A lower distance indicates that the two languages are more similar, potentially facilitating more effective transfer.

Token overlap. We follow (Wu and Dredze, 2019) to measure how many tokens are shared in the source training set and the target test set. A higher token overlap indicates that more tokens were encountered during training in the source language, potentially transferring more supervision from the source to the target.

Development set performance. As small development sets are available in the shared task, we use their performance as an indicator of the transfer performance on test sets, assuming that they share a similar data distribution.[8]

In Figure 4, we show the metric values across different source languages, along with the best source languages identified by distinct metrics. After post-hoc evaluation following the release of test sets, we find that the performance of the development set indeed serves as the most reliable indicator, as the

[8] When training is allowed, it might be more advantageous to use small development sets for training directly rather than source selection, which needs to be further explored.

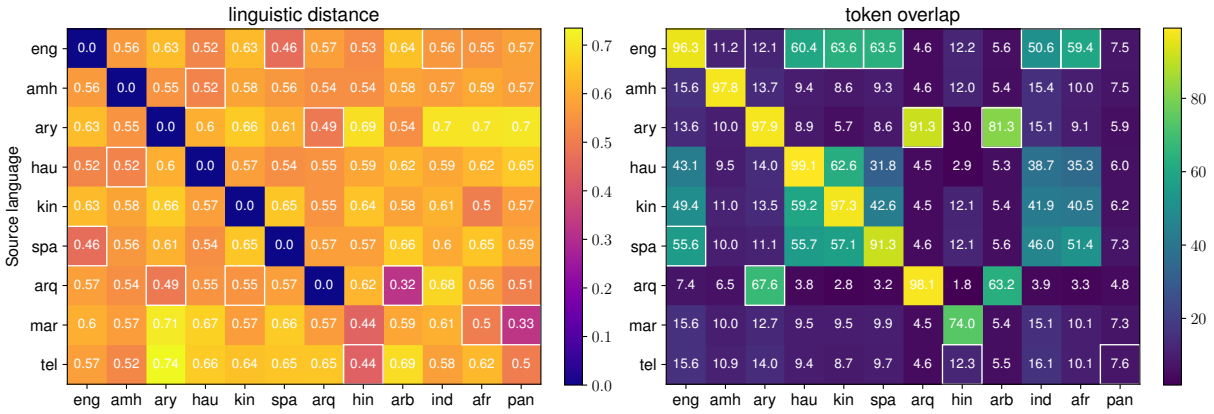
Model	eng	amh	arq	ary	spa	hau	mar	tel	Avg.↑
Baselines w/o training:									
Overlap	56.57	63.28	44.00	**53.76**	58.67	38.86	57.52	60.61	54.16
FastText	55.69	60.64	**44.27**	22.12	57.47	9.19	59.23	69.39	47.25
mpnet-base-v2	**81.94**	69.94	26.35	34.40	56.58	30.86	72.43	56.33	53.60
LaBSE	72.14	**76.49**	40.80	38.58	**63.11**	**41.51**	**73.83**	**75.99**	**60.31**
XLMR-large	39.53	42.07	27.91	4.15	47.59	7.34	40.51	56.36	33.18
AfroXLMR-large	16.55	39.82	20.30	-0.46	30.42	8.13	35.94	30.74	22.68
AfriBERTa-large	53.12	69.23	16.04	13.36	56.68	35.14	20.84	9.73	34.27
AfroXLMR-large-61L	44.10	52.96	32.15	0.35	51.07	17.62	37.66	47.17	35.39
AfroXLMR-large-75L	22.61	37.93	29.38	-2.39	43.58	13.86	32.13	40.42	27.19
Bi-encoders w/ supervised training:									
mpnet-base-v2	85.07	80.43	56.73	75.51	65.29	58.62	81.53	74.49	72.21
LaBSE	84.45	82.59	59.49	78.29	69.02	68.94	**83.97**	76.35	75.39
AfroXLMR-large-61L	82.81	74.61	40.02	66.58	66.65	66.51	38.51	65.73	62.68
Cross-encoders w/ supervised training:									
mpnet-base-v2	80.26	75.04	60.25	80.31	64.92	53.66	65.36	68.54	68.54
LaBSE	86.13	84.75	**60.75**	**82.55**	67.23	69.31	81.10	77.25	**76.13**
AfroXLMR-large-61L	**86.65**	**84.88**	46.61	81.56	**69.08**	**74.65**	75.55	**80.94**	74.99

Table 4: Performance of 10-fold cross-validation on training sets (Spearman's correlation $\times 100$). For each language, we **bold** the best performance achieved in *w/o training* and *w/ supervised training* settings.

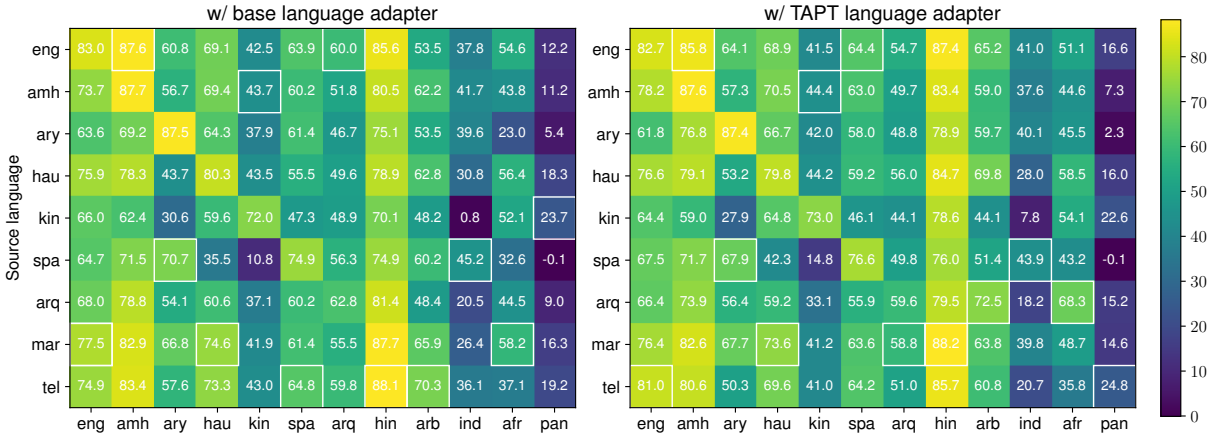
Language	Family / Subfamily	Domain	Corpus Size
English (eng)	Indo-Europoean / Germanic	News, Wikipedia	1.2M
Afrikaans (afr)	Indo-Europoean / Germanic	News, Wikipedia	68k
Amharic (amh)	Afro-Asiatic / Semitic	Community, Wikipedia	250k
Modern Standard Arabic (arb)	Afro-Asiatic / Semitic	News, Wikipedia	110k
Algerian Arabic (arq)	Afro-Asiatic / Semitic	News	244k
Moroccan Arabic (ary)	Afro-Asiatic / Semitic	News	564k
Spanish (spa)	Indo-Europoean / Italic	News, Wikipedia	444k
Hausa (hau)	Afro-Asiatic / Chadic	Community, Wikipedia	564k
Hindi (hin)	Indo-European / Indo-Iranian	News, Wikipedia	472k
Indonesian (ind)	Austronesian / Malayic	News, Wikipedia	92k
Kinyarwanda (kin)	Niger-Congo / Atlantic–Congo	Community	320k
Punjabi (pan)	Indo-European / Indo-Iranian	Wikipedia	412k
Marathi (mar)	Indo-European / Indo-Iranian	News, Wikipedia	856k
Telugu (tel)	Dravidian / South-Central	News, Wikipedia	756k

Table 5: Data statistics for pre-training corpora collected from the Leipzig Corpus Collection.

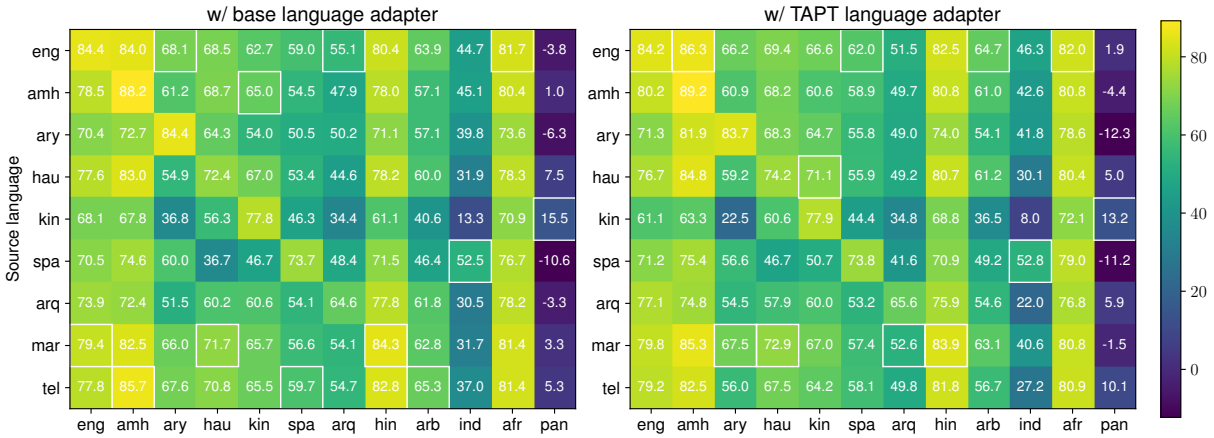
optimal source languages it selected closely align with the ground truth selections.



(a) Left: Linguistic distances between source and target languages. The smallest distance for each target language is highlighted with a box. Right: Token overlaps between source and target languages. The highest overlap for each target language is highlighted with a box. The corresponding source languages are predicted as the best sources for cross-lingual transfer.



(b) Performance on development sets (Spearman's correlation ×100) using different types of language adapters. Boxes are used to highlight the optimal performances for each target language, and the corresponding source languages are predicted as the best sources for cross-lingual transfer.



(c) Performance on test sets (Spearman's correlation ×100) using different types of language adapters. Boxes are used to highlight the optimal performances for each target language, and the corresponding source languages are the ground-truth best sources for cross-lingual transfer.

Figure 4: Comparison of different source language selection methods.

BITS Pilani at SemEval-2024 Task 10: Fine-tuning BERT and Llama 2 for Emotion Recognition in Conversation

Dilip Venkatesh[1], **Pasunti Prasanjith**[1], and **Yashvardhan Sharma**[1]

[1]Birla Institute of Technology and Science, Pilani, Rajasthan, India
Email: {f20201203, pasunti.prasanjith, yash}@pilani.bits-pilani.ac.in

Abstract

Emotion Recognition in Conversation **(ERC)** aims to assign an emotion to a dialogue in a conversation between people. The first subtask of EDiReF shared task aims to assign an emotions to a Hindi-English code mixed conversation. For this, our team proposes a system to identify the emotion based on fine-tuning large language models on the `MaSaC` dataset. For our study we have fine tuned 2 LLMs **BERT** and **Llama 2** to perform sequence classification to identify the emotion of the text.

1 Introduction

Emotion can be defined as a conscious mental reaction subjectively experienced as strong feeling usually directed toward a specific object and typically accompanied by physiological and behavioral changes in the body (Merriam-Webster, 2024). In recent times emotion recognition and sentiment analysis has become increasingly popular due to the research developments in natural language processing. Although similar to sentiment analysis, while sentiment analysis aims to classify text as `POSITIVE`, `NEGATIVE` and `NEUTRAL`, ERC aims to identify text as more in-depth emotions like joy, sadness, anger, contempt etc.

Emotion recognition has multiple use cases in the real world. Opinion mining of conversational data posted by users is done at a large scale at big tech companies. Poria et al. (2019) mentions that ERC has major potential to be used in healthcare systems for psychological analysis and education to understand student frustrations. It is important for language models and chat bots to understand the sentiment of an input text to respond accordingly and generate empathetic dialogue systems (Ma et al., 2020).

For the first subtask of the SemEval 2024 Task 10: *Emotion Discovery and Reasoning its Flip in Conversation (EDiReF)* (Kumar et al., 2024) on CodaLab (Pavao et al., 2023), we aim to conduct Emotion Recognition in Conversation on a Hindi-English code-mixed dataset. Our team proposes a system for this where we fine tune two large language models. Namely the transformer based BERT (Devlin et al., 2019) and Llama 2 (Touvron et al., 2023b).

All of our code can be found on GitHub at `github.com/dipsivenkatesh/SemEval-2024-Task-10`

2 Background

2.1 Task and Data Description

The EDiRef shared task[1] consists of three subtasks.

- Emotion Recognition in Conversation **(ERC)** in Hindi-English code-mixed conversations
- Emotion Flip Reasoning **(EFR)** in Hindi-English code-mixed conversations
- **EFR** in English conversations

In this paper we go through our team's system to solve the first sub task.

The first subtask is to perform ERC on the Hindi-English code-mixed `MaSaC` dataset proposed in Bedi et al. (2023). The dataset comprises of around 1,200 multi-party dialogues from the popular Indian TV show *'Sarabhai vs Sarabhai'*[2] and around 15,000 utterance exchanges (primarily in Hindi) between the speakers. The dataset consisted of the utterances by the speaker and the corresponding emotion label given to each utterance. The emotions were *anger, neutral, contempt, sadness, fear, disgust, joy* and *surprise*.

An example of Emotion recognition in conversation can be found in Table 1

[1]`https://codalab.lisn.upsaclay.fr/competitions/16769`

[2]`https://www.imdb.com/title/tt1518542/`

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 811–815

Speaker	Utterance	Emotion
Sp1	Aaj to bhot awful day tha! (I had an awful day today!)	Sad
Sp2	Oh no! Kya hua? (Oh no! What happened?)	Sad
Sp1	Kisi ne mera sandwich kha liya! (Somebody ate my sandwich!)	Sad
Sp2	Me abhi tumhare liye new bana deti hun! (I can make you a new one right now!)	Joy
Sp1	Wo great hoga! Thanks! (That would be great! Thanks!)	Joy

Table 1: Hindi-English code-mixed conversation with emotions

2.2 Previous Work

Initially the naive Bayes algorithm was used for subject classification (Maron, 1961), specifically for sentiment analysis the variant, binary multinomial naive Bayes algorithm was proposed. More recently, the way to perform classification tasks in natural language processing is through supervised machine learning.

Hazarika et al. (2018b) proposes a conversational memory network (CMN), a method that uses memories to capture inter-speaker dependencies. This was further improved with Interactive Conversational memory Network (ICON) a multimodal method that models the self- and inter-speaker emotional influences into global memories (Hazarika et al., 2018a). The Interaction-Aware Attention Network (IANN) (Yeh et al., 2019) incorporates the contextual information through a novel attention mechanism. It works by by leveraging inter-speaker relation modeling, however it uses distinct memories for each speaker. This is solved with DialougeRNN (Majumder et al., 2019) a method based on RNNs that keeps track of the individual states of speakers throughout conversation. This is then used for emotion classification.

The discovery of Large Language Models (LLMs) have brought in a huge transformation to the field of natural language processing. This is due to to the reasoning and understanding capabilities of these powerful models such as GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023) and LLaMA (Touvron et al., 2023a). Fine tuning of these pre-trained LLMs have showed their versatility and effectiveness across a variety of tasks.

For this task we fine tune 2 models. BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), a model to pre-train bidirectional representations by jointly conditioning on both left and right context in all layers. Due to this, the model can be fine tuned with just one layer to achieve state of the art performance. We also use the Llama 2 7 billion parameter model (Touvron et al., 2023b). We choose the Llama 2 model due to it's state of the art performance on various NLP benchmarks. Due to the large size of Llama 2 we fine tune this model using Parameter Efficient Fine Tuning Methods (Mangrulkar et al., 2022). We do this with Low-Rank Adaptation of Large Language Models (LoRA) (Hu et al., 2021) which freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture. This reduces the number of trainable parameters.

2.3 Evaluation Metrics

The systems used were evaluated with the weighted F1 score metric.

$$\text{Weighted F1} = \sum_{i=1}^{N} \left(\frac{\text{support}_i}{\text{total support}} \right) \cdot \text{F1}_i \quad (1)$$

$$\text{F1}_i = 2 \cdot \frac{\text{precision}_i \cdot \text{recall}_i}{\text{precision}_i + \text{recall}_i} \quad (2)$$

$$\text{where,} \quad \text{precision}_i = \frac{TP_i}{TP_i + FP_i} \quad (3)$$

$$\text{recall}_i = \frac{TP_i}{TP_i + FN_i} \quad (4)$$

and support_i is the number of true instances of class_i and total support is the total number of instances across all classes

3 System Overview

3.1 BERT

We fine-tune the BERT base model (cased) (Devlin et al., 2019) for the emotion classification task with 8 labels. We load the model and train it using the HuggingFace Transformers library (Wolf et al., 2020). The input text is tokenized with the *bert-based-case* tokenizer.

3.1.1 Model Architecture

The model uses the existing BERT base cased architecture. The final layer of the model (the output

layer) is altered to match the 8 classes in the classification task.

3.1.2 Loss Function

For this model we use the **Cross Entropy Loss** between the outputs of the model predictions and the actual labels to optimize the system.

3.2 Llama 2

We fine-tune the Llama 2, 7 billion parameter model (Touvron et al., 2023b) in a similar way in which we fine-tune BERT. We load the model and train it using the HuggingFace Transformers library (Wolf et al., 2020). The input text is tokenized with the *meta-llama/Llama-2-7b-hf* tokenizer.

3.2.1 Model Architecture

Llama 2 model architecture is similar in structure to its predecessor LLaMA (Touvron et al., 2023a) with a context length increase from 2048 to 4096 tokens and usage of Grouped-Query Attention instead of Multi-Query Attention. It is an auto-regressive language model that uses optimized transformer architecture.

3.2.2 Loss Function

We use custom loss function that combines the F1 score and Cross-Entropy Loss to form a single loss value that takes into account both the precision and recall, along with the class imbalances.

4 Experimental Setup

4.1 Dataset Splits

We load the `MaSaC` dataset (Kumar et al., 2023) train, validation and test splits provided to us by the EDiReF shared task organizers using huggingface datasets library (Lhoest et al., 2021). The train set consists of 8506 utterances along with their corresponding label (emotion). The validation set consists of 1354 utterances and the respective label. For final evaluation we are provided with an unlabeled test set of 1580 utterances, to which we must predict the emotion for submission.

4.2 Preprocessing data

Before we pass the inputs to the large language model, we must preprocess the data to an acceptable input format for the large language model, for this we tokenize the datasets.

- **BERT:** For the BERT model we use the pretrained BERT tokenizer *bert-base-cased*. This takes the text of the utterance and generates the *input ids*, *token type ids* and *attention mask*. To make sure all the input sequences have the same length we use maximum length padding. Longer sequences are truncated to the maximum allowable length of the BERT model.

- **Llama 2:** The text for the Llama model is tokenized with the *meta-llama/Llama-2-7b-hf* tokenizer. While tokenizing it is ensured that a space is added before the first token of a given text. The pad token and pad token id are set to the EOS[3] token and EOS token id. While tokenizing, we truncate the longer sequences to the maximum allowable length of the Llama model.

4.3 Training/Fine-tuning

We use the NVIDIA A100 GPUs available on Google Colab for fine-tuning the models.

We load the *bert-base-cased* on HuggingFace for fine-tuning. For the BERT model we use a data loader of batch size 32 while shuffling the data each epoch to not learn any unintended patterns. We use the AdamW optimizer for training (Loshchilov and Hutter, 2019). We set the initial learning rate to be 5×10^{-5} and use a linear learning rate scheduler across the entire duration of training. We then train the model for 4 epochs.

The Llama 2 model is available as *meta-llama/Llama-2-7b-hf* on HuggingFace. We load this model for fine-tuning. Similar to the BERT model, we use a data loader with shuffling for the Llama 2 model, but with a batch size of 16. The AdamW optimzer (Loshchilov and Hutter, 2019) is used while training. Due to the large size of the Llama 2 model, we fine tune the model with PEFT (Mangrulkar et al., 2022) and LoRA (Hu et al., 2021). The LoRA configuration we setup for parameter efficient fine-tuning is as follows. We set the task type as sequence classification, the rank of decompostion matrix (r) is set to 16, the alpha parameter to scale the learned weights (lora alpha) is set to 16 as advised by the LoRA paper. The dropout probability of the LoRA layers is set to 0.05. We do not add any bias term to LoRA layers. We apply LoRA to the projection layers for the query and value components in the attention mechanism of the transformer. We then fine-tune the model for 10 epochs with a learning rate of

[3]End of Speech

1×10^{-4}, warmup ratio of 0.1, maximum gradient norm of 0.3 and a weight decay of 0.001.

5 Results

For evaluation, the organizers rank the system based on weighted F1 score. This is due to the classes being highly imbalanced in the data distribution. The BERT model which was submitted to the leader board achieved a 0.42 weighted F1 score to get 14th place [4]. The performance of all the models can be found in Table 2

	Validation Set	Test Set
BERT	0.43	0.42
Llama 2	0.42	0.41

Table 2: Weighted F1 Scores

Acknowledgements

I would like to thank the organizers of the *EDiReF - SemEval 2024 Task 10* shared task for conducting this competition and organizing this task. I would also like to thank the faculty and research scholars at BITS Pilani for assisting me in my work.

References

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Transactions on Affective Computing*, 14(2):1363–1375.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018a. ICON: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604, Brussels, Belgium. Association for Computational Linguistics.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018b. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2122–2132, New Orleans, Louisiana. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Yukun Ma, Khanh Linh Nguyen, Frank Z. Xing, and Erik Cambria. 2020. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik

[4] https://codalab.lisn.upsaclay.fr/competitions/16769#results

Cambria. 2019. Dialoguernn: An attentive rnn for emotion detection in conversations.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

M. E. Maron. 1961. Automatic indexing: An experimental inquiry. *J. ACM*, 8(3):404–417.

Merriam-Webster. 2024. Emotion. https://www.merriam-webster.com/dictionary/emotion. Accessed: 2024-02-08.

OpenAI. 2023. Gpt-4 technical report.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Sung-Lin Yeh, Yun-Shao Lin, and Chi-Chun Lee. 2019. An interaction-aware attention network for speech emotion recognition in spoken dialogs. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6685–6689.

BITS Pilani at SemEval-2024 Task 9: Prompt Engineering with GPT-4 for Solving Brainteasers

Dilip Venkatesh[1] and **Yashvardhan Sharma**[1]

[1]Birla Institute of Technology and Science, Pilani, Rajasthan, India
{f20201203, yash}@pilani.bits-pilani.ac.in

Abstract

Solving brainteasers is a task that requires complex reasoning prowess. The increase of research in natural language processing has lead to the development of massive large language models with billions (or trillions) of parameters that are able to solve difficult questions due to their advanced reasoning capabilities. The SemEval *BRAINTEASER* shared tasks consists of sentence and word puzzles along with options containing the answer for the puzzle. Our team uses **OpenAI's GPT-4** model along with **prompt engineering** to solve these brainteasers.

1 Introduction

There are two different types of thinking processes, vertical and lateral (Waks, 1997). Vertical thinking refers to the form of linear thinking thinking we are conditioned to. It is based on rationality and logic. Lateral thinking, or *"out-of-the-box"* thinking is a more creative way of thinking from different perspectives. This is contrary to first method.

The recent advancements of natural language processing models, more specifically large language models have achieved great progress in reasoning capabilities and therefore vertical thinking tasks (Talmor et al., 2019, Bisk et al., 2020).

This lateral, creative form of thinking has multiple use cases in the real world since rapid innovation and out of the box thinking are key functionalities of blooming institutions. Innovations are crucial to solve global scale problems like climate change and are very important to big tech companies to keep their consumers happy and engaged. Therefore an interesting part of language models are their abilities to show lateral thinking and defy default commonsense associations.

For the SemEval 2024 Task 9: *BRAINTEASER: A Novel Task Defying Common Sense* (Jiang et al., 2024) on CodaLab (Pavao et al., 2023), we aim to solve the brainteasers as a multiple-choice Question Answering (QA) tasks. Our team proposes a system for this where we use prompt engineering with GPT-4 to solve these brainteasers.

All of our code can be found on GitHub at https://github.com/dipsivenkatesh/SemEval-2024-Task-9

2 Background

2.1 Task and Data Description

The BRAINTEASER shared task[1] consists two different type of brainteasers/puzzles.

- **Sentence Puzzle**: Sentence-type brainteaser where the puzzle defying commonsense is centered on sentence snippets.
- **Word Puzzle**: Word-type brainteaser where the answer violates the default meaning of the word and focuses on the letter composition of the target question

We can find the examples of each puzzle in Table 1. In this paper we go through our team's system to solve both the sentence puzzle and word puzzle task.

The task requires us to to solve the brainteasers in the BRAINTEASER dataset (Jiang et al., 2023). The dataset was created by crawling the internet to find relevant puzzles. This is then filtered to remove irrelevant questions. The task is provided as a question-answering task in which for each puzzle we much select the correct answer from four options.

The task also consists of adversarial subsets to make sure that the approach is based on reasoning and not LLM memorization. The adversarial reconstructions are of two types.

[1]https://codalab.lisn.upsaclay.fr/competitions/15566

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 816–820

Question	Choices
Sentence Puzzle: A man shaves everyday, yet keeps his beard long	He is a barber. He wants to maintain his appearance. He wants his girlfriend to buy him a razor. None of the above.
Word Puzzle: What part of London is in France?	The letter N. The letter O. The letter L. None of the above.

Table 1: Sentence and Word puzzle examples.

- **Semantic Reconstruction** rephrases the original question without changing the correct answer and the distractors.
- **Context Reconstruction** keeps the original reasoning path but changes both the question and the answer to describe a new situational context.

We find instances of adversarial reconstructions in Table 2

2.2 Previous Work

The field of natural language processing has seen massive developments since the discovery of transformers (Vaswani et al., 2023). Initially used in machine translation, transformers found their way into other fields of natural language processing as well including large language models. These large language models like BERT (Devlin et al., 2019), LLaMA/Llama 2 (Touvron et al., 2023a, Touvron et al., 2023b) and OpenAI's GPT-3 (Brown et al., 2020) and GPT-4 (OpenAI, 2023) have powerful reasoning capabilities and can be applied on various tasks involving natural language.

Prompt engineering refers to structuring the input text for a large language model. Methods like prompt engineering and fine-tuning have tremendous efficacy on downstream tasks. If prompted on the role of the language model along with the input question and/or relevant data, language models do a good job on providing the correct output even in a zero-shot manner (Sanh et al., 2022).

There have been quite a few benchmarks for testing the creativity of automatic natural language systems. Identifying puns (Zou and Lu, 2019) and humour (Meaney et al., 2021) is an example of this. The shared task proposed in (Lin et al., 2021) tests the natural language understanding and creativity of it's systems by testing the systems on riddle style questions. This is pretty close to the `BRAINTEASERS` shared task that requires the system to automatically solve brainteasers. The commonsense reasoning ability of these language models are also tested with various benchmarks (Rajani et al., 2019, Ma et al., 2019, Lourie et al., 2021, Maharana and Bansal, 2022). These metrics provide a good analysis of the vertical thinking capabilities of the systems. However for the brainteaser task it is important to think in ways that go against common sense. It is also imperative for the model to understand the questions instead of just memorization as adversarial ways of forming the questions also exist in the task.

2.3 Evaluation Metrics

The systems will be evaluated on their accuracy in the question-accuracy tasks. The following two accuracy metrics are used.

- **Instance-based Accuracy**: where each question individual/adversarial are considered as a separate instance. The accuracy for the original question as well as both of the adversarial ways will be reported.
- **Group-based Accuracy**: This evaluates the accuracy of the original question along with its adversarial reconstructions combined. The value is only counted as correct if it gets all of these questions correct.

3 System Overview

3.1 GPT-4

We use the GPT-4 turbo as `gpt-4-1106-preview` model from the GPT-4 (OpenAI, 2023) family of models. We access the GPT-4 model using the OpenAI API. GPT-4 turbo has a 128,000 token context window and can solve difficult problems with

Adversarial Strategy	Question	Choice
Original	A man shaves everyday, yet keeps his beard long.	He is a barber. He wants to maintain his appearance. He wants his girlfriend to buy him a razor. None of the above.
Semantic Reconstruction	A man preserves a lengthy beard despite shaving every day.	He is a barber. He wants to maintain his appearance. He wants his girlfriend to buy him a razor. None of the above.
Context Reconstruction	Tom attends class every day but doesn't do any homework.	He is a teacher. He is a lazy person. His teacher will not let him fail. None of the above.

Table 2: Adversarial reconstructions of the brainteasers

greater accuracy than previous generation large language models. This is due to its broader general knowledge and advanced reasoning capabilities, its training data is up to the date of April 2023. We use the chat completions API in JSON mode to ensure that we get the correct option answer from the question passed to the model.

3.2 Prompts

We use prompt engineering with the roles of system prompts and user prompts to tell the model what to do and what instructions to follow.

- **Role Prompt**: You are an assistant that only responds in json. You solve riddles and brainteasers that require complex reasoning. Solve the riddle/brainteaser by selecting the correct option from the given option list. The response json should be in the format "optionindex": array index of the option selected from option list. this should be a zero-based index , "optionanswer": The answer selected from the given option list I only want the json output of this.

- **User Prompt**: Solve this brainteaser: (brainteaser question here) optionlist: (answer optionlist here)

With this we can see that we use one role prompt for the entire system, both sentences and word puzzles, and for the user prompt we specify the different questions and the options for the answer.

4 Experimental Setup

We load the BRAINTEASER test datasets (Jiang et al., 2023) provided to us by the BRAINTEASER shared task organizers using the HuggingFace datasets library (Lhoest et al., 2021). For the sentence puzzle we have 120 puzzles with 4 options corresponding to each puzzle and for the word puzzle we have 96 question s and for each question we have 4 options. The test set is unlabeled, it doesn't specify the correct option, and our systems must evaluate the correct option for each brainteaser.

We generate the prompts for each question with the methods specified above and pass them to the GPT-4 turbo chat completions API for solving the brainteasers.

5 Results

For evaluation, the organizers rank the system based on accuracy of the answers on the question-answering task. The GPT-4 with prompt engineering system that we have provided achieves 9th place on the leaderboard in the evaluation phase [2]. The performance of the system on all the different evaluation components can be found in Table 3 for the sentence puzzle and in Table 4 for the word puzzle.

[2] https://codalab.lisn.upsaclay.fr/competitions/15566#results

Team	Original	Semantic	Context	O & S	O & S & C	Overall
GPT-4 + prompt engineering	97.5	92.5	80.0	92.5	77.5	90.0
Human	90.74	90.74	94.44	90.74	88.89	91.98
ChatGPT (zero-shot)	60.77	59.33	67.94	50.72	39.71	62.68
RoBERTa-L	43.54	40.19	46.41	33.01	20.10	43.38

Table 3: Sentence puzzle result.

Team	Original	Semantic	Context	O & S	O & S & C	Overall
GPT-4 + prompt engineering	0.938	0.938	0.875	0.938	0.812	0.917
Human	91.67	91.67	91.67	91.67	89.58	91.67
ChatGPT (zero-shot)	56.10	52.44	51.83	43.90	29.27	53.46
RoBERTa-L	19.51	19.51	23.17	14.63	6.10	20.73

Table 4: Word puzzle result.

Acknowledgements

I would like to thank the organizers of the *SemEval 2024 BRAINTEASER: A Novel Task Defying Common Sense* shared task for conducting this competition and providing us with the data. I would also like to thank the faculty and research scholars at BITS Pilani for assisting me in my work.

References

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 22–32, Hong Kong, China. Association for Computational Linguistics.

Adyasha Maharana and Mohit Bansal. 2022. On curriculum learning for commonsense reasoning. In *Proceedings of the 2022 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 983–992, Seattle, United States. Association for Computational Linguistics.

J. A. Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119, Online. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6:245–255.

Yanyan Zou and Wei Lu. 2019. Joint detection and location of English puns. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2117–2123, Minneapolis, Minnesota. Association for Computational Linguistics.

VHA at SemEval-2024 Task 7: Bridging Numerical Reasoning and Headline Generation for Enhanced Language Models

Harinieswari V[1], Srimathi T[2], Vaishnavi R[3], Aarthi S[4]

Meenakshi Sundararajan Engineering College, Chennai

harinishe22@gmail.com[1], srimathithanasekar@gmail.com[2], vaiiish748@gmail.com[3], aarthigopinath.msec@gmail.com[4]

Abstract

In the dynamic realm of digital media, headline generation stands as a critical force, bridging science and creativity to capture audience interest while ensuring accuracy. Current challenges in numerical integration impede precision, with extractive methods compromising accuracy and abstractive approaches struggling with coherence. Extractive methods, reliant on condensing sentences from source material, often fail to capture nuanced information accurately. Our study pioneers a novel two-step training approach, advancing NLP and emphasizing the crucial need for enhanced numerical reasoning in headline creation. Employing Masked Language Models like BERT and RoBERTa, known for nuanced understanding, and the T5 model's unique text-to-text processing for NLP tasks, our research showcases promising advancements. The Flan-T5 model, integrating external contributions and our dataset, enhances T5's capabilities. Through a rigorous comparative analysis, our study demonstrates the models' effectiveness in overcoming challenges related to numerical integration and headline generation.

1 Introduction

In the dynamic domain of digital media, the synthesis of scientific rigor and creative flair in headline generation is paramount for capturing audience interest while maintaining accuracy. Yet, a persistent challenge arises in integrating numerical data into these headlines with precision. Conventional methods often fall short, either by overlooking crucial numerical insights or sacrificing clarity.Consider the task of distilling information from source material, where existing techniques frequently neglect the nuances of numerical discourse—a critical shortfall, particularly in fields such as finance. This challenge extends to the domain of natural language processing (NLP), where computational systems strive to comprehend and generate human language seamlessly.Our research addresses this challenge through an innovative methodological approach. By leveraging advanced language models like BERT, RoBERTa, and T5, we aim to advance computational linguistics, particularly in reconciling textual narratives with numerical data.Furthermore, we introduce the Flan-T5 model, which integrates external contributions and proprietary datasets to enhance headline generation capabilities. Through systematic comparative analysis, our study validates the efficacy of our approach in overcoming challenges related to numerical integration and headline creation.

NEWS: The US is in the grip of the worst drought in more than 50 years, with almost 80% of the country either in drought or in abnormally dry conditions. The NOAA's latest report finds that 56% of the continental US is in drought, the sixth-highest percentage on record and the worst since 1956, reports the Washington Post. Topsoil has dried out and crops, pastures, and rangeland have deteriorated at a rate rarely seen in the last 18 years, the NOAA says. The Department of Agriculture has declared the drought the biggest disaster in its history, and forecasters expect little relief in the short term for the middle of the country, where corn and soybean crops have been devastated. I have never seen this type of weather before like this. A lot of old timers haven't either, a farmer in Kansas who has seen his corn crop wither and his cattle pastures dry up tells the AP. I just think we are seeing history in the making.

DistilRoBERTa :"Drought Reaches Unprecedented Levels in the US, Worst Since 1956 - NOAA Report"

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 821–828

FLAN T5: "Unprecedented Drought Grips US, Surpassing 1956 Record, NOAA Report Reveals"
T5: "NOAA: US Facing Worst Drought Since 1956, Agriculture Department Declares Historic Disaster"

Table 1: Sample Data for Headline Generatiion

In essence, our work underscores the importance of advancing computational methodologies to reconcile textual and numerical information. By doing so, we not only refine headline generation practices but also contribute to broader discussions on information dissemination in the digital era, fostering enhanced engagement and understanding among diverse audiences.

2 Related work

2.1 Graph-based Neural Networks

Shuzhi[1] proposed a paper on Fake News Detection through Graph-based Neural Networks provides a detailed examination of techniques, focusing primarily on graph-based methodologies.This system lacks a comprehensive comparative analysis with empirical validation across diverse approaches and datasets.

2.2 Seq2seq Model

Khairul[2] paper introduces a Multitasking-Based Seq2seq Model, SEQ2SEQ++, aiming to enhance chatbot performance. While comparing with two recent models, It lacks a comprehensive analysis against a wider range of existing techniques.

2.3 LaMini-LM

Abdul[3] paper proposes LaMini-LM, a technique to create smaller models from instruction-tuned large language models (LLMs) to address resource-intensive issues. LaMini-LM achieves comparable performance to strong baselines through meticulous fine-tuning and a diverse set of instructions. This approach optimizes resource utilization, making it suitable for resource-constrained environments.It lacks in generalizing the large models and different architectures due to less scalable performance and less applicability across settings.

2.4 NumNet:

Qiu Ran[10] paper introduces NumNet, a numerical machine reading comprehension (MRC) model employing a numerically-aware graph neural network for improved numerical reasoning. This models becomes complex for higher mathematical operations and computation costs are high during training.

3 Dataset Description

3.1 Subtask 1: Fill the Blank In News Headline

The NumHG dataset, consisting of 21,157 news stories from Newser, forms the basis for Subtask 1 by concealing numbers within masked headlines. The organized validation set of 2,572 articles follows a structured approach, featuring four columns: "News" (article content), "Masked Headline" (hiding numbers), "Calculation" (operations, copy, round, paraphrase, and conversion), and "Answer" (correct numerical values). This methodical structure serves as a robust foundation for constructing and evaluating models, facilitating the task of filling in blank news headlines with hidden numbers.

3.2 Subtask 2: Headline Generation

The dataset for Subtask 2 includes 2,365 validation and 21,157 training news articles. Differing from Subtask 1, this subset prioritizes headline creation over filling blank spaces, omitting the "calculation" column. The dataset structure is meticulously curated for cohesive training, sharing headlines with Subtask 1 articles for a unified approach. This strategic curation enhances overall dataset continuity and reliability for our study project.

4 Methodology

4.1 Proposed Models

4.1.1 Masked Language Model

Masked Language Models, exemplified by BERT, predict masked tokens in sentences like news headlines. RoBERTa, a more advanced version, improves upon BERT's design with enhanced linguistic pattern recognition. Trained on a dataset over 10 times larger than BERT, RoBERTa excels in discerning subtle nuances. Its dynamic masking strategy during training boosts its ability to acquire robust word representations.

DistilRoBERTa offers a streamlined, efficient alternative without sacrificing essential features.

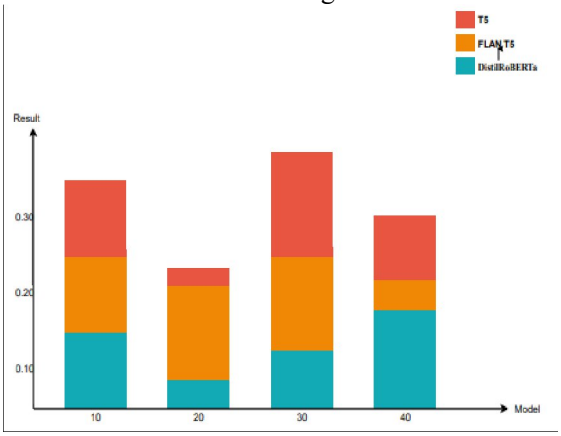


Figure 1:Result Vs Models for Subtask 1

4.1.2 T5 Language Model

The T5 model, or Text-to-Text Transfer Transformer, employs a unique approach in processing text input and generating corresponding text output for various NLP tasks. Unlike BERT, T5 utilizes a method introduced by Mishra in 2020, replacing consecutive tokens with a single "Mask" keyword. Specifically tailored for tasks like text summarization and headline generation, T5 diverges from BERT's focus on predicting individual words. In our research, we leverage external contributions, including Michal Pleban's training of the T5-base model on a dataset of 500k articles with headings, aimed at generating concise headlines (Pleban, 2020). Caleb Zearing's significant efforts in training T5 on a large collection of Medium articles for generating article titles also contribute to our research (Zearing, 2022). Building upon both Pleban's and Zearing's models, we enhance training with our proprietary dataset to advance NLP capabilities further.

4.1.3 Flan-T5 Model

The Text-to-Text Transfer Transformer (T5) offers a unique approach to handling text input and generating equivalent text output in various NLP applications. Unlike BERT, T5 utilizes a single "Mask" keyword to replace multiple consecutive tokens, as introduced by Mishra in 2020, enhancing its capability for tasks like text summarization and headline creation. Building upon T5's framework, we incorporate models trained for specific tasks by external researchers, such as T5-base-en-generate-headline (Pleban, 2020), designed for generating concise headlines from articles.

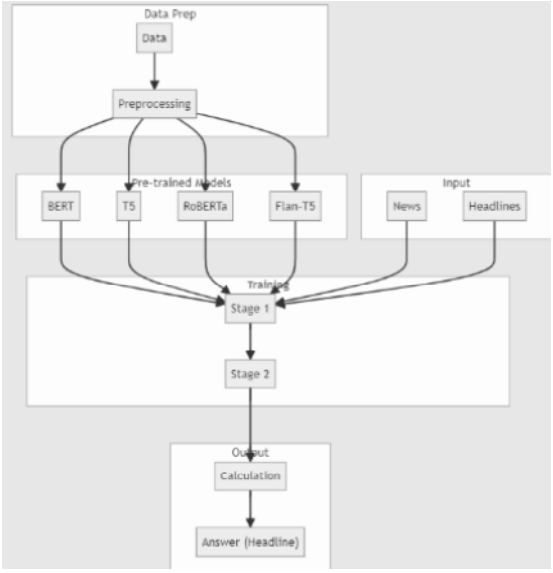


Figure 3: Proposed Architecture

Our research aims to leverage the valuable contributions of external researchers like Michal Pleban, expanding the understanding of T5's versatility in diverse applications.

4.2 Subtask 1

4.2.1 DistilRoBERTa

In order to prepare the dataset for DistilRoBERTa training, we combined pertinent columns and replaced underscores in the headlines with mask tokens. We used input-output pairs to train the model with a learning rate of 5e-5. To improve the model's predictive power, we gave the top 20 vocabulary tokens for numerical value extraction during training priority. Our objective was to improve DistilRoBERTa's numerical reasoning task performance by means of meticulous optimization and sophisticated training methods. This thorough method guarantees accurate and contextually relevant output, improving the model's usefulness in headline generation and other NLP tasks.

4.2.2 T5 & Flan-T5 Models - Train in One Step

We expanded training by including two additional T5-based models alongside Flan-T5. For masked headlines, we replaced underscores with a token and combined them with news columns as inputs, excluding the calculation column due to its

negative impact on performance. Flan-T5 was trained with a learning rate of 2e-5, while T5 models used 5e-5. A method to extract numerical values for blanks was implemented by finding the token index in each headline. Our aim was to enhance the models' accuracy in generating numerical values in headlines through iterative refinement of training settings, emphasizing the importance of adapting training approaches to optimize performance in tasks like numerical reasoning and headline generation.

4.2.3 T5 & Flan-T5 Models - Train Twice in Two Steps

The training procedure for T5 and Flan-T5 models involved two phases aimed at enhancing prediction accuracy and comprehension. Initially, the models were trained using news and masked headlines, with the calculation column as labels to understand the relationship between headlines, news content, and calculations.

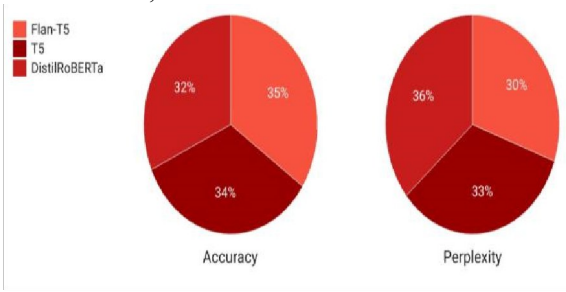


Figure 2: Accuracy Vs Perplexity for Subtask 1

In the second phase, the models were trained with the answer column as output and the calculation column as input to reinforce comprehension of calculation methods. This systematic approach ensured precise headline creation. Flan-T5, built upon the T5 architecture, revolutionizes text processing for NLP tasks by replacing successive tokens with a single "Mask" term, improving performance in tasks like text summarization and headline creation. By leveraging expertise from models like T5-base-en-generate-headline (Pleban, 2020), T5 becomes more versatile across applications, thanks to contributions from researchers like Michal Pleban.

4.3 Subtask 2

Based on T5 architecture, the Flan-T5 model transforms text production and handling for NLP applications. In contrast to BERT, it replaces successive tokens with a single "Mask" term, improving performance in tasks like text summary and headline creation. By incorporating models that are experts at creating succinct headlines, such as T5-base-en-generate-headline (Pleban, 2020), we increase the utility of T5. The excellent contributions of outside researchers such as Michal Pleban have allowed T5 to become more versatile in a wider range of applications.

5 Result

Subtask 1: Fill the Blank In News Headline

In our comprehensive assessment of seven distinct models—Czearing, Czearing with Two Steps, Lamini, Lamini with Two Steps, Michau, Michau with Two Steps, and DistilRoBERTa-based—our primary metric for evaluation was perplexity.

Model	***Accuracy (%)***	***Perplexity (Before)***	***Perplexity (After)***
Czearing (Single Step)	85.7	3.21	1.45
Czearing (Two Steps)	82.4	3.45	1.58
Flan-T5 (Single Step)	88.9	2.66	1.05
Flan-T5 (Two Steps)	90.2	2.18	0.92
Michau/t5-base	86.5	2.89	1.12
DistilRoBERTa-base	78.3	4.75	2.39

Table 2: Model Perplexity Before and After Training

The results showcased a notable enhancement in performance across all models post-training, indicating improved proficiency in numerical reasoning tasks.

Flan-T5 (Two Steps) emerged as the top performer in accuracy, boasting an impressive 90.2%. This model exhibited exceptional competence in arithmetic operations, decimal rounding, and handling complex mathematical operations.

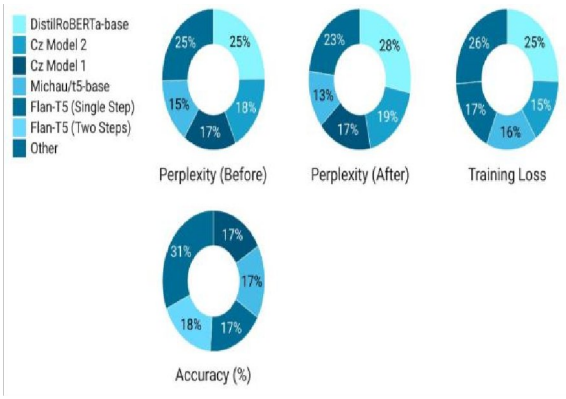


Figure 4:Performance Metrics for Subtask 1

A detailed analysis of error patterns revealed specific challenges encountered by the models, particularly in the domains of arithmetic operations, rounding decimal numbers, and combinations of various mathematical operations. These insights provide valuable guidance for refining the models and addressing their limitations.

The Czearing models demonstrated competitive performance, with the base Czearing achieving a training loss of 0.0250. Notably, Czearing with Two Steps exhibited comparable results, indicating the efficacy of the two-step approach.

news	masked headline	calculation	ans	headline
As of Jan. 1, Walmart will no longer offer 30...	<extra_id_0> K Walmart Part-Timers to Lose Hea...	Paraphrase(30,000,K)	30	30K Walmart Part-Timers to Lose Health Insurance
Dax Shepard and Kristen Bell got married at t...	Dax Shepard: Wedding to Kristen Bell Cost $<ex...	Copy(142)	142	Dax Shepard: Wedding to Kristen Bell Cost $142
Nancy Reagan, the helpmate, backstage adviser...	Nancy Reagan Dead at <extra_id_0> </s>	Copy(94)	94	Nancy Reagan Dead at 94
American Airlines faces FAA fines of more tha...	American Airlines Faces $<extra_id_0> M Fine f...	Copy(7)	7	American Airlines Faces $7M Fine for Safety Vi...
Ingrid Lyne, the Seattle mom allegedly murder...	$<extra_id_0> K Raised for Kids of Mom Dismemb...	Paraphrase(222,000,K)	222	$222K Raised for Kids of Mom Dismembered on Date

Figure 5: Sample Data for Subtask 1 using Czearing (one step) model

MBZUAI/LaMini-Flan-T5-783M models showcased effective headline generation, achieving a training loss of 0.1411 and a validation loss of 0.1869 over four epochs. This performance underscores the model's proficiency in numerical reasoning tasks.

T5-based models, such as Michau/t5-base-en-generate-headline, demonstrated a significant reduction in perplexity from 2.66 to 1.05, showcasing enhanced numerical reasoning capabilities. The DistilRoBERTa-based model (distilroberta-base) also displayed successful adaptation to numerical reasoning, with perplexity decreasing from 6.23 to 3.68.

Our comparative analysis reveals that both T5-based and DistilRoBERTa-based models exhibit promising performance in numerical reasoning tasks. Particularly, the Flan-T5 model, especially in its Two Steps variant, stands out with superior accuracy in subtask 1. These findings provide valuable insights into the effectiveness and versatility of transformer-based models in addressing complex numerical reasoning applications. The observed improvements in perplexity post-training underscore the adaptability and learning capabilities of these models in handling diverse numerical challenges.

Error Type	***Examples***
Arithmetic Operations	Misinterpretation of mathematical symbols
Rounding Decimals	Incorrect rounding of numerical values
Combination of Operations	Challenges in handling complex expressions

Table 3: Error Patterns for Subtask 1

Subtask 2: Headline Generation

The first model, czearing/article-title-generator, harnessed the T5-base architecture during a 10-epoch training phase. This process yielded promising results with a training loss of 1.3876 and a validation loss of 1.6684. The tokenization methodology involved a maximum sequence length of 2024 for input and 128 for labels.

Figure 6: Comparision between T5 and Flan T5 Model for Subtask 2

Our evaluation process included a meticulous analysis of headline predictions using the ROUGE-L metric. The model demonstrated a proficiency in generating headlines that are not only contextually relevant but also exhibit a

nuanced understanding of numerals. To illustrate, when faced with the news snippet "US Soldier Held After Killing 5 at Baghdad Base," the model's prediction, "US Soldier Charged With Killing 5 at Stress Clinic," received a commendable ROUGE-L score of 0.74. A similar success was observed with the news piece "Nintendo Chief Dies at 55," where the model predicted "Nintendo President Dead at 55" with an impressive ROUGE-L score of 0.92.

Moving to the second model, michau/t5-base-en-generate-headline, which employed the T5-base-en-generate-headline architecture, underwent 7 epochs, achieving a training loss of 1.3329 and a validation loss of 1.6855. Tokenization parameters included a maximum sequence length of 2024 for input and 256 for labels.

In terms of predictions, this model also displayed competitive performance, albeit with a different focus. The ROUGE-L scores reflected the model's proficiency in numeral-aware headline generation. For instance, when presented with the news snippet "3 Killed in California Quarry Shooting Spree," the model predicted "3rd Victim Dead in Quarry Shooting; Manhunt St..." and obtained a ROUGE-L score of 0.38. Similarly, for the news piece "Dow Up 305 on Election Day," the predicted headline "Stocks Up 305 in Election Rally" garnered a ROUGE-L score of 0.50.

Both models exhibited noteworthy capabilities in capturing not only the essence of the news but also the specific nuances associated with numerals. The competitive ROUGE-L scores across different samples affirm the models' efficacy. These results suggest a potential application of these models in real-world scenarios where numeral-aware headline generation is crucial. The nuanced understanding of numerals showcased by these models positions them as valuable assets in the evolving landscape of natural language processing tasks.

5 Conclusion

Our research presents a significant stride in advancing numerical reasoning within the domain of news headline generation. The thorough evaluation of transformer models, including Flan-T5, DistilRoBERTa, and T5 variants, showcased remarkable improvements in accuracy for filling blank headlines with hidden numbers. Flan-T5 (Two Steps) particularly stood out with a commendable 90.2% accuracy, demonstrating exceptional competence in arithmetic operations and handling complex mathematical expressions. Additionally, the nuanced understanding of numerals displayed by T5 models in Subtask 2 underscores their efficacy in generating contextually relevant headlines. These findings collectively contribute valuable insights into the evolving landscape of natural language processing, especially in tasks involving numerical reasoning and headline creation.

References

[1]Shuzhi Gong, Richard O. Sinnott, Jianzhong Qi The University of Melbourne, Melbourne, VIC 3000, Australia-2023. *Fake News Detection through Graph-based Neural Networks.*

[2] Kulothunkan Palasundram,Nurfadhlina Mohd Sharef , Khairul Azhar Kasmiran , and Azreen Azman, (Member, IEEE)-2021. *SEQ2SEQ++: A Multitasking-Based Seq2seq Model to Generate Meaningful and Relevant Answers*

[3] Minghao Wu1,2∗Abdul Waheed1 Chiyu Zhang1,3 Muhammad Abdul-Mageed1,3 Alham Fikri Aji1 -2024. LaMini-LM: A Diverse Herd of Distilled Models from Large-Scale Instructions.

[4] Mingye Wang 1,*, Pan Xie 1 , Yao Du 1 and Xiaohui Hu2-2023.*T5-Based Model for Abstractive Summarization: A Semi-Supervised Learning Approach with Consistency Loss Functions.*

[5] Colin Raffel∗ craffel,Noam Shazeer∗, Adam Roberts∗, Katherine Lee∗ ,Sharan Narang s ,Michael Matena Yanqi Zhou Wei Li ,Peter J. Liu-2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Alfred. V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling, volume 1*. Prentice-Hall, Englewood Cliffs, NJ.

[6] Antonio Mastropaolo∗ , Simone Scalabrino† , Nathan Cooper‡ , David Nader Palacio‡ , Denys Poshyvanyk‡ , Rocco Oliveto† , Gabriele Bavota-2021. *Studying the Usage of Text-To-Text Transfer Transformer to Support Code-Related Tasks.*

[7] Hyung Won Chung, Le Hou, Shayne Longpre, Bar ret Zoph, Yi Tay Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex an Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. *Scaling instruction finetuned language models*

.

[8]Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. *Injecting numerical reasoning skills into language models.* In Proceedings of the 58th Annual Meeting of the Association for Computational

Linguistics, pages 946–958, Online. Association for Computational Linguistics.

[9] Dominic Petrak, Nafise Sadat Moosavi, and Iryna Gurevych. 2023. *Arithmetic-based pre training improving numeracy of pretrained language models*. In Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023), pages 477–493, Toronto, Canada. Association for Computational Linguistics.

[10] Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: *Machine reading comprehension with numerical reasoning.* In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2474–2484, Hong Kong, China. Association for Computational Linguistics.

[11] Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024).

A Appendices

A. Model Architecture

Model	Architecture Details
DistilRoBERTa	Trained on masked headlines with top 20 vocabulary tokens
T5 Language Model	Incorporates external contributions for diverse applications
Flan-T5 Model	Built upon T5 architecture, enhancing text summarization

Table A.1: Model Architecture Details

B. Dataset Overview

Dataset Component	Composition
NumHG (Subtask 1)	21,157 news stories from Newser
Headline Generation (Subtask 2)	2,365 validation, 21,157 training news articles

Table B.1: Dataset Summary

C. Training Approach

Model	Training Approach
DistilRoBERTa	Input-output pairs with calculation column as labels
T5 & Flan-T5	One-step and two-step training for enhanced accuracy

Table C.1: Training Approaches

D. Evaluation Metrics

Subtask	Metric	Noteworthy Achievements
Subtask 1	Perplexity	DistilRoBERTa: Enhanced numerical reasoning
Subtask 2	ROUGE-L Scores	czearing/gen-title: Contextually relevant headlines

Table D.1: Evaluation Metrics

E. Language and Library Used

Package	Version	Usage
Pandas	1.3.3	Data manipulation and analysis
Matplotlib	3.4.3	Data visualization
Seaborn	0.11.2	Statistical data visualization
NLTK	3.6.2	Natural Language Processing (NLP) tasks
Scikit-learn	0.24.2	Machine learning models and metrics
TensorFlow	2.6.0	Deep learning framework for model development

Keras	2.6.0	High-level neural networks API (TensorFlow backend)
Joblib	1.0.1	Parallel computing library for Python
Statsmodels	0.12.2	Statistical models and tests
Requests	2.26.0	HTTP library for making API requests

Table E.1: Packages Used for the Experiment

TueSents at SemEval-2024 Task 8: Predicting the Shift from Human Authorship to Machine-generated Output in a Mixed Text

Valentin Pickard and Hoa Do
Seminar für Sprachwissenschaft
Eberhard Karls Universität Tübingen, Germany
{valentin.pickard, hoa.do}@student.uni-tuebingen.de

Abstract

This paper describes our approach and results for the SemEval 2024 task of identifying the token index in a mixed text where a switch from human authorship to machine-generated text occurs. We explore two BiLSTMs, one over sentence feature vectors to predict the index of the sentence containing such a change and another over character embeddings of the text. As sentence features, we compute token count, mean token length, standard deviation of token length, counts for punctuation and space characters, various readability scores, word frequency class and word part-of-speech class counts for each sentence. class counts. The evaluation is performed on mean absolute error (MAE) between predicted and actual boundary word index. While our competition results were notably below the baseline, there may still be useful aspects to our approach.

1 Introduction

With the rapid proliferation of Large Language Models (LLMs) that are able to produce fluent texts in response to user queries across a wide range of domains and topics, concerns are raised about the potential misuses of such powerful tools. In spite of their fluency, LLM-generated texts may contain factual errors, inadvertently spreading misinformation. Another common issue occurs in the education system, where students may attempt to pass off the responses of such an LLM as their own work, evading commonly used safeguards against plagiarism. Given the overwhelming volume of potentially machine-generated content, it is desirable to have automated means of detecting such texts to address the above-mentioned issues. In this task (Wang et al., 2024), we examine exclusively English mixed texts, where a switch from human authorship to LLM output occurs at most once in a text sample (some samples are entirely machine-generated). To us, this models a plausible use case, where a human user employs an LLM to finish their work for them. For each sample, the task is to predict the token index at which the authorship change occurs. We observe, that due to the structure of the samples, we can reformulate the task more generally as trying to detect an authorship change and its location within the sample texts, without explicitly trying to detect the presence of LLM-generated text. This allows us to adapt more traditional, computationally relatively inexpensive approaches to stylometry and authorship identification/attribution. While the task is formulated as prediction of a boundary word, we begin by identifying the boundary sentence in which the authorship change occurs. For each sentence, textual feature vectors are extracted and combined with character n-gram information, those sentence vectors are then fed into a Bidirectional LSTM network (Hochreiter and Schmidhuber, 1997) which is trained to predict the boundary sentence. We found that our approach performed reasonably well in-domain on the development set, in spite of inevitably introducing some token offset error by only making sentence level predictions and choosing the middle tokens, but failed out-of-domain on the test set, ranking at 26 out of 30 in the competition on subtask C.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 829–832

2 Background

In accordance with the task guidelines we do not use external data, but use the English subsets of larger data sets from subtask A and B to extract a character vocabulary. The subtask C dataset comprises a bit over 4000 texts with 505 pre-split into a dev set by the task authors, each text labeled with the index of the boundary word. The following table shows token and sentence counts for train and dev set, when tokenized by splitting on whitespace (U+0020) as in the task baseline model. Sentence splits are determined subsequently on the token lists by identifying sentence-final tokens using our detection regex, this is done to ensure matching the given boundary word labels.

	train	dev
texts	3,649	505
sentences	41,570	5,628
tokens (types)	864,153 (29,593)	116,221 (8,641)
chars	5,933,701	803,771
avg. sentences	11.4	11.1
avg. boundary	3.4	3.4

Table 1: Task data statistics

We observe that about half of the samples contain 4 - 11 sentences and sentence count per sample ranges from 1 to 76, with 24 samples containing just one sentence, like e.g. *"We have added a 2+ page* ***discussion*** *on the experimental results, highlighting the superiority of the ARC-based models and their impact on the field of deep learning."* (boundary word 'discussion' in bold). While on average the author switch occurs in the fourth sentence, in about 15-20% of the samples the switch occurs in the first sentence. Examining the boundary word position within their respective sentences we found an average offset of -1.6 (train set) or -1.8 (dev set) from the middle of the sentence, i.e. the switch occurs slightly before mid sentence on average.

An example text, split in sentences and tokens can be observed below, with the boundary word **"baseline"** at index 20 highlighted:

- Format: *label*: (*start token index*) [*tokens*]
- 0: (0) ['The', 'paper', 'proposes', 'a', 'method', 'to', 'recognize', 'time', 'expressions', 'from', 'text.']
- 1: (11) ['It', 'is', 'a\r\nsimple', 'rule-based', 'method,', 'which', 'is', 'a', 'strong', '**baseline**', 'for', 'time', 'expression', 'recognition.']
- 0: (25) ['The', 'authors', 'analyze', 'different', 'datasets', 'and', 'discover', 'that', 'only', 'a', 'small', 'set', 'of', 'words', 'are', 'consistently', 'used', 'to', 'convey', 'time', 'information.']
- remaining sentences omitted for brevity

Interestingly, by using the given tokenization method on whitespace only, we preserve linebreak characters such as in the third token of the second sentence, and obtain empty string tokens in between multiple whitespaces. We choose to include both this 'raw' text data as well as a normalized version in our system, since on the one hand, such typographic choices are indicative of authorship changes but on the other hand, we may not be able to rely on their presence in unseen data.

3 System overview

In our system, we at first sought to compare and combine more traditional textual features with task-specific learned character embeddings, as the former offer the benefit of cheap computation and greater transparency, whereas the latter should allow the system to capture more subtle patterns at the expense of transparency and at a higher computational cost. We choose a relatively straightforward basic architecture for our models, using a BiLSTM over sentence vectors to predict the boundary sentence at which the authorship change occurs. With regards to textual features, we compute for each sentence: token count, mean token length, standard deviation of token length, counts for punctuation and

space characters, various readability scores, word frequency class and word part-of-speech class counts. For our character model, we used another BiLSTM with embedding layer over the text characters, adjusting the token labels to the character level. As the character level model did not perform to our expectations on both normalized and raw text, we did not combine it with the textual feature model and decided to use the latter as a stand-alone model.

4 Experimental setup

We used the provided train/dev split to tune our models. We compared performances of a purely textual feature based model and a character-based model. We extracted the textual features offline, using the *spacy* (Honnibal and Montani, 2017) library for Python and its *textdescriptives* (Hansen et al., 2023) extension library. We used the *en_core_web_sm (v3.7.1)* pipeline for spacy. Hyperparameters were tuned manually, we settled on single-layer networks of hidden size 16, using Adam optimizer (Diederik, 2014) with learning rate 1e-5, training on batches of size 8 over 100 epochs. For the character-based network we choose an embedding size of 8. The task is evaluated on mean absolute error (MAE) between predicted and actual boundary word index. To translate our boundary sentence prediction into a token index, we selected the middle token index as default, rounding it down for sentences with an even token count. For the character model, we chose the token containing the predicted character index.

5 Results

On the development set our textual feature model showed somewhat promising results with regards to predicting the sentence containing the boundary word. It predicted the correct sentence in 69.9% of cases, the adjacent sentence in a further 21.4% with a sentence index MAE of 0.47. Translating these predictions into token level predictions using the sentence mid-point yielded a token MAE of 13.5, notably worse than the baseline model's. Our character embedding model did perform notably worse on the development set, with a token MAE of 48.9. We therefore did not pursue it further and abandoned our initial idea of combining it with the textual feature model. On the test set, for the competition, we submitted the predictions of our textual feature model, unfortunately not matching the performance on the development set. We only managed to predict 30% of boundary sentences correctly, with another 22.6% predictions of the adjacent sentence, resulting in a sentence MAE of 3.2 and a disappointing token MAE of 59, ranking 26 out of 30 among the participant models in subtask C. We suspect this drop in performance mainly be caused by introduction of a new text domain in the test set, while development and training samples are exclusively drawn from PeerRead (Kang et al., 2018) i.e. academic peer reviews, the test set introduces student essays from OUTFOX (Koike et al., 2023). This degradation of performance for stylistic textual features is in line with other findings, e.g. the comparisons performed on the M4 dataset (Wang et al., 2023) of which this competition's dataset is an extension.

6 Conclusion

While our model's performance leaves plenty of room for improvement, we can envision the use of simple textual features in a lightweight model, similar to ours as a basic tool in contexts where more powerful models are either unavailable or too expensive to run and the text domain is known in advance. Focusing on sentence level instead of token level predictions allows us to reduce computational effort and we consider it sufficient for many practical applications, where automated detection of LLM generated text is only a first step, such as e.g. examining student essays, where we would expect a teacher to follow up with affected students regarding suspicious spans of text individually.

References

P. K. Diederik. 2014. Adam: a method for stochastic optimization.

Lasse Hansen, Ludvig Renbo Olsen, and Kenneth Enevoldsen. 2023. Textdescriptives: A Python package for calculating a large variety of metrics from text. *Journal of Open Source Software*, 8(84):5153.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. Outfox: LLM-generated essay detection through in-context learning with adversarially generated examples.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. SemEval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

TECHSSN1 at SemEval-2024 Task 10: Emotion Classification in Hindi-English Code-Mixed Dialogue using Transformer-based Models

Venkatasai Ojus Yenumulapalli, Pooja Premnath, Parthiban Mohankumar, Rajalakshmi Sivanaiah and Angel Deborah Suseelan
Department of Computer Science and Engineering,
Sri Sivasubramaniya Nadar College of Engineering,
Chennai - 603110, Tamil Nadu, India
{venkatasai2110272, pooja2110152, parthiban2110207}@ssn.edu.in,
{rajalakshmis, angeldeborahs}@ssn.edu.in

Abstract

The increase in the popularity of code mixed languages has resulted in the need to engineer language models for the same . Unlike pure languages, code-mixed languages lack clear grammatical structures, leading to ambiguous sentence constructions. This ambiguity presents significant challenges for natural language processing tasks, including syntactic parsing, word sense disambiguation, and language identification. This paper focuses on emotion recognition of conversations in Hinglish, a mix of Hindi and English, as part of Task 10 of SemEval 2024. The proposed approach explores the usage of standard machine learning models like SVM, MNB and RF, and also BERT-based models for Hindi-English code-mixed data- namely, HingBERT, Hing mBERT and HingRoBERTa for subtask A.

1 Introduction

Code-mixed Hindi and English, also referred to as 'Hinglish', has gained widespread usage, especially in the realm of social media. With the increasing prevalence of code-mixed languages like Hinglish, there arises a necessity to analyze and understand this linguistic material. While language models designed for individual languages like English or Hindi (Ly, 2022) are quite robust and effective, they often struggle to perform well with code-mixed languages. This difficulty stems from the colloquial nature of the conversations in code-mixed dialogue, with no formal grammar rules.

Traditional machine learning models perform well on code-mixed data only when the nature of the classification task is simple, like in the form of sentiment analysis (classification into positive, neutral, and negative emotions). Task 10 of SemEval 2024 (Kumar et al., 2024) contains emotions from the extended Ekman model(Ekman, 1992), which contain emotions that are more complex to discern and distinguish between like contempt versus anger.

This paper explores the usage of both classical machine learning models as well as Transformer-based BERT models, specifically designed for Hinglish data.

2 Related Work

Thakur et al. (2020) delve into the current landscape of Hindi-English code-mixed natural language processing and their work meticulously surveys the progress made in sentiment analysis within this domain while also dissecting the inherent issues and challenges it encounters.

Sentiment analysis in code-mixed data is done in a plethora of ways, spanning from machine translation to corpus processing based on sentence structure. Jadhav et al. (2022) introduced a framework employing a pipeline for the conversion of Hinglish to English, offering a structured approach to the task. Similarly, Sinha and Thakur (2005) present a method for translating Hinglish to both English and Hindi, leveraging Hindi and English morphological analyzers and implementing cross-morphological analysis to achieve accurate conversion. Ensemble learning for identifying emotions in contextual texts was proposed by (Angel Deborah et al., 2020). Additionally, (S et al., 2022) proposed a lexicon-based solution for recognising emotions in Tamil texts.

Das and Singh (2023) embraced a deep learning paradigm, implementing convolutional neural networks (CNN), long short-term memory (LSTM), and bi-directional long short-term memory (Bi-LSTM) for sentiment analysis. Meanwhile, Ravi and Ravi (2016) conclusively identified a combination of TF-IDF vectorizer, gain ratio-based feature selection, and a Radial Basis Function Neural Network (RBFN) as the optimal pipeline for sentiment analysis of Hinglish data. Patwa et al. (2020) utilized M-BERT and the Transformers framework,

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 833–838

diverging from traditional methods. Singh (2021) employed diverse techniques for sentiment analysis of Hinglish, leveraging various embeddings such as count vectorizer and word2vec across different machine learning algorithms including SVM, KNN, and Decision Trees. A similar work by (Deborah et al., 2022) focused on recognizing emotions using Gaussian Process and decision trees.

However, the task of emotion classification poses a much greater challenge compared to the simpler task of sentiment analysis. It necessitates the utilization of specific techniques to process and balance data across a broader spectrum of classes. This paper attempts to utilize both traditional and Transformer based approaches for Hinglish emotion classification.

3 Dataset

The SemEval 2024 Task 10 dataset (Kumar et al., 2023) comprises 8056 samples, featuring fields such as ID, speaker, utterance, and emotion. The ID uniquely identifies each episode of the conversation, while the speaker field denotes the person speaking. The utterance field represents the dialogue, expressed in Hinglish, and the emotion field indicates the corresponding emotion conveyed in the utterance. Adding on, the validation dataset contains 1354 samples while the test dataset contains 1580 samples. Table 1 shows the distribution of labels in the dataset.

Emotion	Count
Anger	819
Contempt	542
Disgust	127
Fear	514
Joy	1596
Neutral	3909
Sadness	558
Surprise	441

Table 1: Distribution of emotions and their respective counts.

4 Data Preprocessing

In the domain of code-mixed emotion recognition, preprocessing the utterances is essential for effective model training. The emotion column, representing a spectrum of eight distinct emotions—'disgust', 'contempt', 'anger', 'neutral', 'sadness', 'fear', and 'surprise'—is encoded using a label encoder for standardized representation. Code-mixed data inherently presents spelling ambiguities, demanding robust normalization techniques. For example, the word 'friend' in Hindi could be spelled as 'dost', 'dhosth', 'dhost' etc. Spelling correction is done using a phonetic similarity assessment. For each word, a phonetic code is computed and identifies feasible correction candidates from a dynamically created phonetic dictionary. The Levenshtein distance metric is used to evaluate the dissimilarity between the input word and potential corrections. This procedure is applied to all the utterances, on each word. The resultant corrected words are subsequently merged to form a spell-corrected utterance. A dictionary of all the speakers is also created, and the speaker names present in the utterances are removed, along with numbers and symbols.

5 Proposed Methodology

5.1 Support Vector Machine, Multinomial Naive Bayes and Random Forest

To classify the utterances into one of the eight emotion classes, emotion labels were encoded using LabelEncoder. The CountVectorizer transformed text into numerical features. Initially, standard classification models like Support Vector Machines (SVM), Multinomial Naive Bayes (MNB), and Random Forest(RF) were utilized. These models were chosen based on their suitability for text classification tasks and their potential effectiveness in handling emotion classification within Hindi-English code-mixed data. These models were trained on the training set and evaluated on the validation set using accuracy and the weighted F1 score metrics. Table 2 and 3 shows the precision scores and other performance metrics of each of the standard machine learning models.

5.2 Long Short Term Memory (LSTM)

A Bidirectional LSTM model was then leveraged to address the challenges that could not be resolved by the SVM, MNB, and RF models. This model architecture is well-suited for sequential data processing tasks due to its inherent ability to capture long-range dependencies in text sequences. Figure 1 shows the architecture diagram of the Bidirectional LSTM model.

This bidirectional processing allows the model to effectively capture contextual information from

Emotion	SVM	MNB	RF
Anger	0.00	0.12	**0.19**
Contempt	**0.33**	0.00	0.17
Disgust	0.00	0.00	**1.00**
Fear	**0.33**	0.00	0.24
Joy	0.55	**0.58**	0.55
Neutral	0.43	0.43	**0.44**
Sadness	0.00	0.27	**0.28**
Surprise	0.22	**0.29**	0.27

Table 2: Precision scores of standard machine learning models

Metric	SVM	MNB	RF
Testing Accuracy	**0.44**	0.40	0.43
Testing Weighted F1	0.31	0.30	**0.33**

Table 3: Performance metrics of standard machine learning models

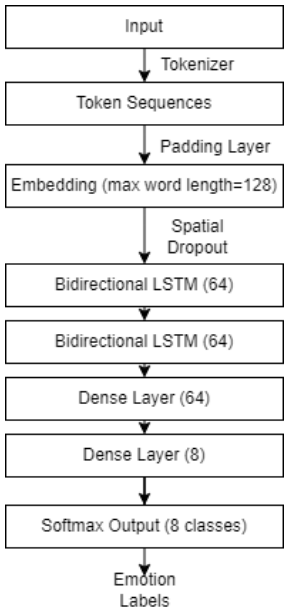


Figure 1: Architecture diagram of the Bidirectional LSTM model

preceding and succeeding words. The model architecture is described as follows:

Embedding Layer: This layer transforms input words into dense vectors of fixed size. It facilitates the representation of words in a continuous vector space, where similar words have similar representations.

Spatial Dropout1D Layer: This layer applies dropout to the input features with a dropout rate of 0.2. It helps prevent overfitting by randomly dropping input units during training.

Bidirectional LSTM Layers: The model consists of two Bidirectional LSTM layers. Each layer comprises 64 units and processes input sequences in both forward and backward directions.

Dense Layers: Two dense layers follow the LSTM layers. The first dense layer has 64 units and uses the ReLU activation function. The final dense layer has 8 units (equal to the number of emotion classes) and uses the softmax activation function for multi-class classification.

The training parameters are as follows:

Optimizer: The model is optimized using the Adam optimizer, a popular choice for training neural networks due to its adaptive learning rate.

Loss Function: Sparse categorical cross-entropy is used as the loss function, suitable for multi-class classification tasks with integer-encoded target labels.

Early Stopping: Training includes early stopping with a patience of 3 epochs. It monitors the loss metric and restores the best weights when no improvement is observed after the specified number of epochs.

Emotion	LSTM Precision Values
Anger	0.06
Contempt	0.08
Disgust	0.017
Fear	0.48
Joy	0.38
Neutral	0.12
Sadness	0.12
Surprise	0.21

Table 4: Precision scores of LSTM model

Batch Size: Training is performed with a batch size of 32.

Epochs: The model is trained for a maximum of 10 epochs.

The Bidirectional LSTM model achieves a test accuracy of **0.35** with a weighted F1 score of **0.43** on the testing set. Table 4 shows the precision scores of LSTM model.

5.3 Hindi-English Code Mixed BERT Models

The usage of BERT (Bidirectional Encoder Representations from Transformers) models tailored for Hindi-English code-mixed data can significantly enhance the accuracy and effectiveness of emotion classification tasks. These models are pre-trained on large corpora of code-mixed text and can be fine-tuned for specific classification tasks. In this

section, three models from the L3Cube Pune team (Nayak and Joshi, 2022), are utilized- namely HingBERT, Hing-mBERT, and HingRoBERTa.

5.3.1 HingBERT

HingBERT, akin to its BERT counterpart, comprises a stack of transformer blocks, typically 12 in number, with self-attention mechanisms and feed-forward neural networks. The model's architecture includes special tokens such as [CLS] and [SEP] to denote sentence boundaries and separation.

5.3.2 Hing mBERT

Hing mBERT inherits the architecture of BERT but is trained across a multitude of languages, including Hindi and English. Its architecture remains consistent with BERT's stack of transformer blocks, each equipped with self-attention mechanisms for capturing contextual information.

5.3.3 Hing RoBERTa

Hing RoBERTa, an extension of the RoBERTa architecture, delves into the intricacies of Hindi-English code-mixed text by integrating advanced architectural modifications. Built upon the foundation of RoBERTa's transformer-based architecture, Hing RoBERTa leverages deeper stacks of transformer layers, intricate attention mechanisms, and optimized weight initialization strategies to handle the nuances of bilingual conversations. With augmented batch sizes and increased learning rates, Hing RoBERTa optimizes gradient descent algorithms to navigate the vast parameter space effectively(Liu et al., 2019). Figure 2 shows the architecture diagram of the Transformer-based models.

5.3.4 Implementation

The implemented framework revolves around fine-tuning the HingBERT, Hing mBERT, and Hing RoBERTa Transformer-based models.

Architecture: The architecture is characterized by the transformer's ability to capture long-range dependencies and intricate contextual nuances within text sequences. Each model comprises a series of transformer blocks, with HingBERT and Hing mBERT featuring 12 transformer layers, while HingRoBERTa encompasses a more extensive architecture with 12 or more layers, as per its pre-defined configuration. Within each transformer block, self-attention mechanisms enable the model to dynamically weigh the importance of individual tokens based on their contextual relevance, facilitating effective feature extraction and representation learning.

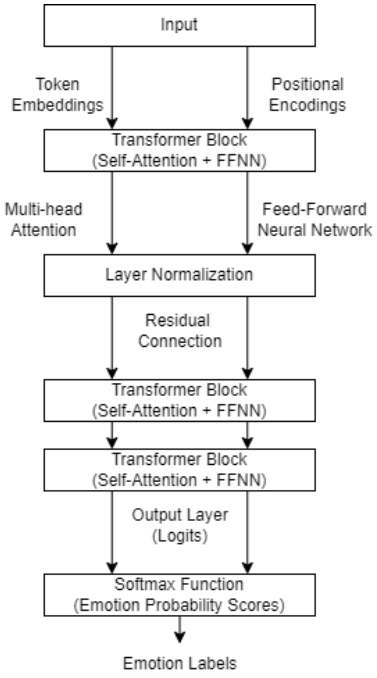


Figure 2: Architecture diagram of Transformer-based models

Multi-Head Attention Mechanism: The attention mechanism, a pivotal component of the transformer architecture, is augmented with multi-head attention, allowing the model to attend to different parts of the input sequence simultaneously.

Feed-Forward Neural Networks (FFNN): Following the self-attention mechanism, token representations are fed through feed-forward neural networks (FFNN) within each transformer block. FFNNs consist of multiple layers of linear transformations, interspersed with non-linear activation functions, such as the Rectified Linear Unit (ReLU), facilitating nonlinear transformations and feature extraction at each layer.

Gradient Clipping: Gradient clipping is employed during the backpropagation phase to alleviate the issue of exploding gradients, ensuring stable training dynamics and promoting convergence.

Embedding Layers: Token embeddings are employed to represent individual tokens within the input sequences, with dimensions determined by the pre-trained embedding matrices. Positional encodings are added to the token embeddings to convey positional information, allowing the model to differentiate between tokens based on their relative positions within the sequence.

Emotion	Hing BERT	Hing mBERT	Hing RoBERTa
Anger	0.28	0.27	**0.33**
Contempt	0.19	0.16	**0.26**
Disgust	**0.25**	0.20	0.20
Fear	0.24	0.23	**0.34**
Joy	0.45	0.49	**0.54**
Neutral	**0.52**	**0.52**	**0.52**
Sadness	0.35	0.28	**0.36**
Surprise	0.31	**0.34**	0.30

Table 5: Precision scores of BERT based models

	Hing BERT	Hing mBERT	Hing RoBERTa
Accuracy	0.45	0.44	**0.47**
Weighted F1	0.42	0.43	**0.45**

Table 6: Performance metrics of BERT based models

Activation Functions and Layer Normalization: Activation functions such as the GELU (Gaussian Error Linear Unit) are applied within the feed-forward neural networks to introduce non-linearity and enable the modeling of complex relationships within the data.

Tables 5 and 6 show the precision value across emotions and the accuracy and weighted F1-scores for the three Transformer-based models.

6 Results and Analysis

6.1 SVM, MNB and RF

The Support Vector Machine (SVM) classifier demonstrates varying performance across different emotions. Notably, it achieves relatively high precision for Contempt and Fear classes, scoring 0.33 for each. However, its precision is very low for Anger, Disgust, and Sadness, achieving 0.00 precision for these emotions. SVM's performance seems to struggle particularly with emotions characterized by intensity and subtlety. Multinomial Naive Bayes (MNB) exhibits competitive performance, particularly evident in its precision for Joy and Surprise emotions, achieving 0.58 and 0.29 respectively, which are among the highest precision values across all models.

Random Forest (RF) emerges as a robust performer across various emotions, demonstrating balanced precision values across the emotion spectrum. RF achieves perfect precision (1.00) for Disgust, indicating its capability to discern this emotion accurately within code-mixed text. Additionally, RF performs consistently well for Neutral and Sadness emotions.

While SVM and MNB show specific strengths for certain emotions, such as Fear and Joy respectively, RF emerges as a more balanced performer across the emotion spectrum, particularly excelling in capturing nuances associated with Disgust.

6.2 LSTM

The LSTM model's precision values exhibit notable variations across different emotions. While it achieves relatively high precision in classifying Fear (0.48) and Joy (0.38), its performance significantly diminishes in categorizing Disgust (0.017) and Anger (0.06). Despite its recurrent nature and ability to retain sequential information, the LSTM model appears to struggle with the contextual intricacies present in the emotion classification task. It achieves a weighted F1-score of 0.43.

6.3 Hindi-English Code-Mixed BERT Models

The BERT-based models showcase more consistent and generally higher precision values across various emotions. Specifically, Hing RoBERTa emerges as the top performer among the BERT-based models, achieving the highest precision scores in several emotional categories, including Contempt (0.26), Fear (0.34), Joy (0.54), and Sadness (0.36). Hing BERT and Hing mBERT also demonstrate competitive precision values, albeit slightly lower than Hing RoBERTa. HingRoBERTa achieves the highest weighted F1-score of 0.45. Table 5 and 6 shows the precision scores and other performance metrics of BERT-based models.

Our team, TechSSN1, placed 7th out of 39 participating teams in the shared subtask A.

7 Conclusion

The future scope of this work entails improving and enhancing the proposed models to handle a wider variety of data. The unstructured nature of Hinglish poses a challenge to the model's performance. By understanding the nuances, fine-tuning can be implemented to enhance the model's efficacy. Additionally, the work can be extended to encompass the classification of other types of emotions apart from the traditional Ekman model and refined to undertake tasks such as sarcasm or humor detection.

References

S Angel Deborah, S Rajalakshmi, S Milton Rajendram, and TT Mirnalinee. 2020. Contextual emotion detection in text using ensemble learning. In *Emerging Trends in Computing and Expert Technology*, pages 1179–1186. Springer.

Shubham Das and Tanya Singh. 2023. Sentiment recognition of hinglish code mixed data using deep learning models based approach. In *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 265–269. IEEE.

S Angel Deborah, Rajendram S Milton, TT Mirnalinee, and S Rajalakshmi. 2022. Contextual emotion detection on text using gaussian process and tree based classifiers. *Intelligent Data Analysis*, 26(1):119–132.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Ishali Jadhav, Aditi Kanade, Vishesh Waghmare, Sahej Singh Chandok, and Ashwini Jarali. 2022. Code-mixed hinglish to english language translation framework. In *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, pages 684–688. IEEE.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Cong Khanh Ly. 2022. English as a global language: An exploration of efl learners' beliefs in vietnam. *International Journal of TESOL Education*, 3:19–33.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas Pykl, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. *arXiv preprint arXiv:2008.04277*.

Kumar Ravi and Vadlamani Ravi. 2016. Sentiment classification of hinglish text. In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pages 641–645. IEEE.

Varsini S, Kirthanna Rajan, Angel S, Rajalakshmi Sivanaiah, Sakaya Milton Rajendram, and Mirnalinee T T. 2022. Varsini_and_Kirthanna@DravidianLangTech-ACL2022-emotional analysis in Tamil. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 165–169, Dublin, Ireland. Association for Computational Linguistics.

Gaurav Singh. 2021. Sentiment analysis of code-mixed social media text (hinglish). *arXiv preprint arXiv:2102.12149*.

R Mahesh K Sinha and Anil Thakur. 2005. Machine translation of bi-lingual hindi-english (hinglish) text. In *Proceedings of Machine Translation Summit X: Papers*, pages 149–156.

Varsha Thakur, Roshani Sahu, and Somya Omer. 2020. Current state of hinglish text sentiment analysis. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*.

SHROOM-INDElab at SemEval-2024 Task 6: Zero- and Few-Shot LLM-Based Classification for Hallucination Detection

Bradley P. Allen, Fina Polat and **Paul Groth**
University of Amsterdam
Amsterdam, NL
{b.p.allen, f.yilmazpolat, p.t.groth}@uva.nl

Abstract

We describe the University of Amsterdam Intelligent Data Engineering Lab team's entry for the SemEval-2024 Task 6 competition. The SHROOM-INDElab system builds on previous work on using prompt programming and in-context learning with large language models (LLMs) to build classifiers for hallucination detection, and extends that work through the incorporation of context-specific definition of task, role, and target concept, and automated generation of examples for use in a few-shot prompting approach. The resulting system achieved fourth-best and sixth-best performance in the model-agnostic track and model-aware tracks for Task 6, respectively, and evaluation using the validation sets showed that the system's classification decisions were consistent with those of the crowd-sourced human labellers. We further found that a zero-shot approach provided better accuracy than a few-shot approach using automatically generated examples. Code for the system described in this paper is available on Github[1].

1 Introduction

Prompt engineering of large language models (LLMs) (Liu et al., 2023) has recently emerged as a viable approach to the automation of a wide range of natural language processing tasks. Recent work (Allen, 2023) has focused on the development of zero-shot chain-of-thought (Wei et al., 2022; Kojima et al., 2022) classifiers, where hallucination in generated rationales is a concern. Hallucination detection (Ji et al., 2023; Huang et al., 2023) is a way to determine whether the outputs of such systems are sensible, factually correct and faithful to the provided input. The SemEval-2024 Task 6 (Mickus et al., 2024) allows us to evaluate whether and how applying techniques we have developed in the above mentioned work and with related work on knowledge extraction (Polat et al., 2024) using zero- and few-shot classification can provide a means of addressing this concern. Previous systems that perform prompt engineering of LLMs as a means to implement hallucination detection include SelfCheckGPT (Manakul et al., 2023) and ChainPoll (Friel and Sanyal, 2023).

2 Data and Task

The challenge provides a dataset consisting of data points containing: the specific task that a given language model is to perform; an input given to the language model on which to perform that task; a target that is an example of an acceptable output, and the output produced by the language model. Table 1 shows an example of such a data point.

Task	Definition Modeling
Input text	"The Dutch would sometimes <define> inundate </define> the land to hinder the Spanish army ."
Target text	"To cover with large amounts of water; to flood."
Generated text	"(transitive) To fill with water."

Table 1: Example data point from the unlabeled training dataset for the model-agnostic task.

Hallucination detection is framed as a binary classification task, where the classifier assigns either 'Hallucination' or 'Not Hallucination' labels with associated probability estimates to data points. Classifier performance is evaluated by comparing these assignments and probabilities to human judgments and their probability estimates, using accuracy and Spearman's correlation coefficient (ρ) for assessment. Around 200 crowd-sourced human labellers each labeled about 20 data points. The competition features two tracks: model-agnostic, which uses the basic setup, and model-aware, adding a field for the Hugging Face model identifier of the model generating the text for each data point. Each track provides an unlabeled training dataset and labeled validation and test datasets.

[1] https://www.github.com/bradleypallen/shroom/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 839–844

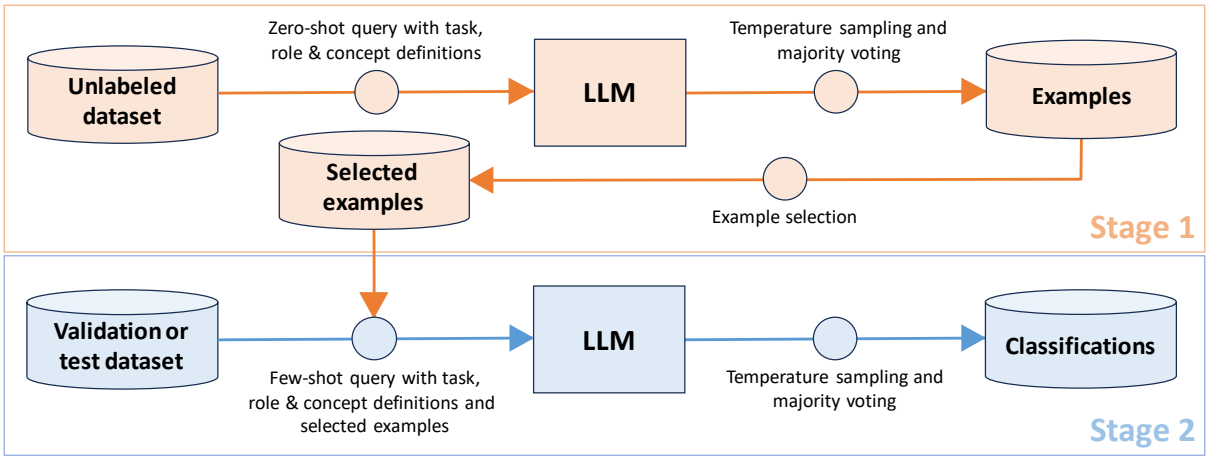


Figure 1: SHROOM-INDElab system workflow.

3 Approach

Our submission for the SHROOM task is a system that defines classifiers for hallucination detection using prompt engineering of an LLM. Figure 1 shows the two-stage workflow used to produce the classifier and evaluate it using the SHROOM datasets.

In Stage 1, we use in-context learning where we ask the LLM to perform the classification according to provided task, role, and concept definition in a zero-shot manner without providing any examples. These classified data points provide examples for a few-shot classifier used in Stage 2. We now proceed to describe the query design and processing steps in the workflow.

3.1 Zero- and few-shot query design

Figure 2 provides an example of the query used to prompt an LLM to produce a classification. The basic prompt template consists of instructions on how to evaluate the generated text according to a hallucination concept definition to answer the question if the generated text is a hallucination or not. Specific guidance is provided such that the form of the answer is in the labels needed to compare directly to the label test data.

The task associated with the data point determines the context for generating both the zero shot and the few shot query based on the prompt template, as illustrated in Figure 2. For the zero-shot query, no examples are included.

The elements involved in instantiating the template given a data point include the task definition performed by another LLM to produce the generated text, a role definition that we assign the classifier to perform, and the concept definition that frames hallucination phenomena and criteria to consider an output as hallucination. The use of role play with LLMs is described by (Shanahan et al., 2023) and its use in the context of zero-shot reasoning is described in (Kong et al., 2023). The role definition describes a persona that the LLM is instructed to assume in the context of making a classification decision. For example, for the Definition Modeling task, we instruct the LLM to assume the persona of a lexicographer. The task and role definitions for each task are shown in Table 2. We also provide a single concept definition for the notion of hallucination that is held constant across all of the tasks.

Task	Task definition	Role definition
Definition Modeling (DM)	The given task is Definition Modeling, meaning that the goal of the language model is to generate a definition for a specific term in the input text.	You are a lexicographer concerned that the generated text accurately captures the meaning of the term between the '<define>' and '</define>' delimiters in the input text.
Paraphrase Generation (PG)	The given task is Paraphrase Generation, meaning that the goal of the language model is to generate a paraphrase of the input text.	You are an author concerned that the generated text is an accurate paraphrase that does not distort the meaning of the input text.
Machine Translation (MT)	The given task is Machine Translation, meaning that the goal of the language model is to generate a natural language translation of the input text.	You are a translator concerned that the generated text is a good and accurate translation of the input text.
Text Simplification (TS)	The given task is Text Simplification, meaning that the goal of the language model is to generate a simplified version of the input text.	You are an editor concerned that the generated text is short, simple, and has the same meaning as the input text.

Table 2: Task and role definitions used for in-context learning.

3.2 Temperature sampling and majority voting

Part of the task involves producing an estimate of the probability that a data point exhibits hallucination. In the SHROOM-INDELab system, the estimated probability is calculated by performing temperature sampling (Ackley et al., 1985), query-

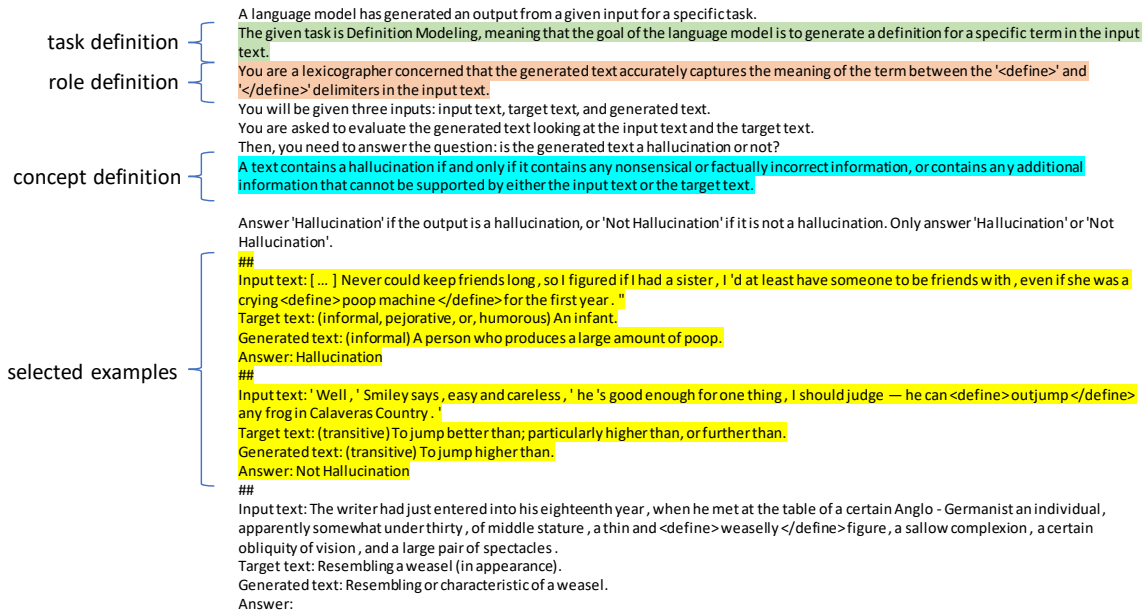


Figure 2: Example prompt for a Stage 2 classifier, given a Definition Modeling task data point from one of the SHROOM datasets, and using 1 example per label.

ing the LLM multiple times to generate a sample of classifications, and then dividing the number of positive classifications (i.e., where the generated label is 'Hallucination') by the total number of classifications in the sample. Temperature sampling is performed in producing both Stage 1 zero-shot and Stage 2 few-shot classifications.

3.3 Example selection

In Stage 1, the algorithm processes an unlabeled dataset to generate examples using a zero-shot query. Following the Self-Adaptive Prompting approach described in (Wan et al., 2023a,b), for each task type we sample 64 data points from the unlabelled dataset, and then use a zero-shot query to obtain a classification with estimated probability of hallucination. This information is combined with the data point to produce an example. We partition the examples per task type into two pools, one with positive examples where the label is 'Hallucination' and the other with negative examples where the label is 'Not Hallucination'.

The process used to select the examples to include in the prompt is shown in Algorithm 1. The first example chosen from each pool is the one with the maximum negative entropy of the classification probability, as defined in Equation 1:

$$F_0(p) = p * \log p + (1-p) * \log(1-p) \quad (1)$$

Algorithm 1 Select examples given a task and label

Require: P: generated examples for given task and label, K: number of selections
Ensure: S: selected examples

$S \leftarrow \emptyset$
$Pool \leftarrow P$
for $k \leftarrow 0$ **to** $K-1$ **do**
  if $k == 0$ **then**
    $s_k \leftarrow \arg\max_{p \in Pool} F_0(p)$
  else
    $s_k \leftarrow \arg\max_{p \in Pool} F(p, S)$
  end if
  $S \leftarrow S \cup \{s_k\}$
  $Pool \leftarrow Pool \setminus \{s_k\}$
end for

For each remaining selection $i \leq K$, the algorithm selects the example that maximizes a trade-off between the diversity of prompts and the consistency of the majority voting result, as defined in Equation 2:

$$F(p, S) = F_0(p) - \lambda \cdot \max_{s \in S}(1 - \text{sim}(\phi(p), \phi(s))) \quad (2)$$

ϕ is calculated for a given data point by concatenating its data into a string and then using an embedding model to produce a representation vector. This trade-off is quantified by subtracting a weighted maximum cosine similarity of the embeddings from the negative entropy, with the weight λ controlling the balance between diversity and consistency. In all of our experiments, in keeping with

(Wan et al., 2023b), λ is set to 0.2. The selected examples for both labels are then serialized and concatenated. This concatenated string is then used to augment the zero-shot query prompt given the task.

4 Experimental Setup and Results

The LLMs used in the evaluating the system were from OpenAI (`gpt-3.5-turbo`, `gpt-4-0125-preview`) and were invoked using the OpenAI API with the LangChain Python library. Stage 1 was performed once with $K = 5$ using `gpt-4-0125-preview` on 25 January 2024. The embedding model used in the calculation of ϕ was OpenAI `text-embedding-ada-002`. The Stage 2 run for our final submission during the evaluation period was conducted on 28 January 2024. Runs for the hyperparameter and ablation study results reported below were conducted between 17 February 2024 and 18 February 2024. Approximately $500 USD in OpenAI API charges were incurred during the above runs.

4.1 Classification performance

As shown in Table 3, using `gpt-4-0125-preview` and `gpt-3.5-turbo` as LLMs our approach showed a significant improvement in both accuracy and Spearman's ρ over the baseline reported for the model-agnostic and model-aware validation sets.[2]

Our best-performing submission to the competition used `gpt-4-0125-preview` as its LLM with 1 example provided per label, 20 samples for majority voting, and a temperature setting of 1.2. We compare it to the baseline system's performance on the test datasets together with that reported for each of the first ranked teams in the model-agnostic track (GroupCheckGPT) and the model-aware track (HaRMoNEE). The SHROOM-INDElab system ranked fourth and sixth in the tracks, respectively.

The values of ρ can be interpreted as showing a moderate to strong correlation between the estimated probability of hallucination provided by the system and that provided by the majority vote result of the human labellers.

[2]Although we submitted results for the model-aware track, our implementation of the approach is model agnostic and does not utilize the model field of the data point.

4.2 Hyperparameter study

The classifier has three hyperparameters; temperature, which is the parameter passed to the language model to indicate the level of stochasticity associated with its generation process; the number of examples per label provided for in-context learning; and the number of samples per query performed and used to calculate the estimated probability associated with the classification of the data point.

We investigated the impact of varying the values of the three hyperparameters of the classifier on the classifier's performance. We used `gpt-3.5-turbo` to conduct this investigation, computing values of accuracy and Spearman's ρ by executing three different passes over the model-agnostic validation dataset.

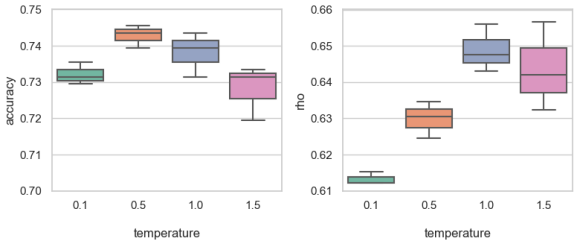


Figure 3: Classifier performance by temperature.

Figure 3 shows the best classifier accuracy is obtained with a temperature between 0.5 and 1.0, and that the best value for Spearman's ρ is obtained with a temperature between 0.5 and 1.5, given settings of 1 example per label and 5 samples per query.

Figure 4 shows that increasing the number of examples for few-shot classification beyond one per label led to an increase in accuracy with diminishing returns after 2 examples per label, but a decrease in Spearman's ρ, given settings of temperature of 1.0 and 5 samples per query.

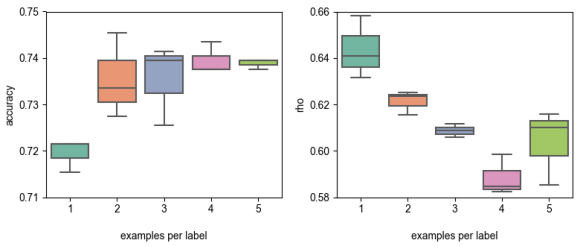


Figure 4: Classifier performance by examples per label.

Figure 5 shows that increasing the number of samples per query led to an increase in both accuracy and Spearman's ρ, given 1 example per label

Dataset	System	model-agnostic accuracy	model-agnostic ρ	model-aware accuracy	model-aware ρ
Validation	Baseline	0.649 (+0.000)	0.380 (+0.000)	0.707 (+0.000)	0.461 (+0.000)
	SHROOM-INDElab (`gpt-3.5-turbo`)	0.773 (+0.124)	0.652 (+0.272)	0.764 (+0.057)	0.605 (+0.144)
	SHROOM-INDElab (`gpt-4-0125-preview`)	**0.814** (+0.165)	**0.697** (+0.317)	**0.772** (+0.065)	**0.635** (+0.174)
Test	Baseline	0.697 (+0.000)	0.403 (+0.000)	0.745 (+0.000)	0.488 (+0.000)
	SHROOM-INDElab (`gpt-4-0125-preview`)	0.829 (+0.132)	0.652 (+0.249)	0.802 (+0.057)	0.605 (+0.117)
	HaRMoNEE	0.814 (+0.117)	0.626 (+0.223)	**0.813** (+0.068)	0.699 (+0.210)
	GroupCheckGPT	**0.847** (+0.150)	**0.769** (+0.366)	0.806 (+0.061)	**0.715** (+0.227)

Table 3: Classifier performance on SHROOM datasets. ρ = Spearman's correlation coefficient.

and a temperature of 1.0.

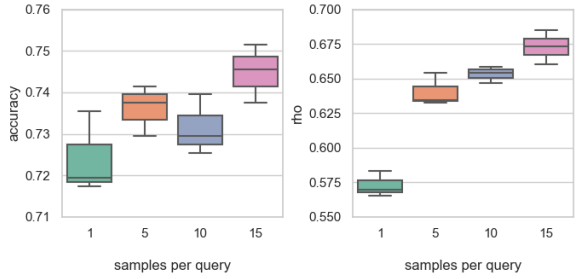


Figure 5: Classifier performance by samples per query.

4.3 Ablation study

Figure 6 shows the results of an ablation study to determine the contribution of the various elements of the prompt provided to the language models. We evaluated the contribution of each of the components of the Stage 2 classifier prompt by removing each in sequence, in the following order: the selected examples, the task definition, the role definition, and finally the concept definition. The ablation study was conducted using `gpt-3.5-turbo`, with 1 example per label, 5 samples per query, and a temperature of 1.0, again involving three different passes over the model-agnostic validation dataset.

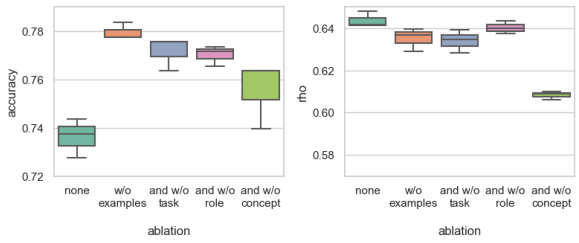


Figure 6: Ablation study using the model-agnostic validation dataset.

We interpret the results of the ablation study as indicating that the use of examples led to poorer accuracy but slightly better Spearman's ρ, that the contributions of the definitions of task and role towards classifier performance were minimal, but that the contribution of the definition of the concept of hallucination was significant.

4.4 Level of agreement with human labellers

We also investigated the degree of inter-annotator alignment exhibited with respect to the model-agnostic data set. Based on the human labeling data associated with each data point in the model-agnostic validation data set, we obtained a Fleiss' κ of 0.373, which can be interpreted as indicating a fair level of agreement among the human labellers, which in turn implies that the reliability of the human labeling might be reasonable, but is not highly consistent or unanimous. Adding the classifier's labeling yields an increase in Fleiss' κ to 0.405, closer to a moderate level of agreement, which implies that the classifier's decisions are consistent with those of the human labellers.

human consensus	N	accuracy	κ	ρ
low (2/3 split)	145	0.621	0.238	0.224
high (4/5 split)	171	0.854	0.701	0.734
unanimous	183	**0.929**	**0.856**	**0.885**
all	499	0.814	0.623	0.697

Table 4: Alignment between the system and human labellers.

We then proceeded to investigate the relationship between the degree of agreement between human labellers and system performance. Table 4 shows the level of agreement between the system and the human labellers, as measured by taking subsets of data points from the model-agnostic validation dataset filtered by the three degrees of consistency in human labeling and calculating the pairwise Cohen's κ between the system's labeling and the label provided by taking the majority vote of the human labellers. The results indicate that system agreement with human labeling increases as the certainty of the human labeling increases.

5 Discussion and Conclusion

In summary, the SHROOM-INDElab system was competitive with the other systems submitted for evaluation, and system labeling was consistent with that of human labellers.

The result in the ablation study that the exclusion of selected examples led to better accuracy suggests the need for further investigation with respect to how the way in which examples are selected and included in the classifier prompts impacts accuracy to determine the cause of the problem. The result that the exclusion of an explicit definition of hallucination leads to poorer accuracy and Spearman's ρ suggests the utility of including intentional definitions of concepts in prompts for LLM-based classifiers (Allen, 2023).

Given the above results, we plan to investigate the use of this approach to hallucination detection in future work on the evaluation of natural language rationale generation (Li et al., 2024) in the context of zero- and few-shot chain-of-thought classifiers for use in knowledge graph evaluation and refinement (Allen et al., 2023).

Acknowledgements

This work is partially supported by the European Union's Horizon Europe research and innovation programme within the ENEXA project (grant Agreement no. 101070305).

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169.

Bradley P Allen. 2023. Conceptual engineering using large language models. *arXiv preprint arXiv:2312.03749*.

Bradley P. Allen, Lise Stork, and Paul Groth. 2023. Knowledge Engineering Using Large Language Models. *Transactions on Graph Data and Knowledge*, 1(1):3:1–3:19.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *arXiv preprint arXiv:2310.18344*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xin Zhou. 2023. Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702*.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. 2024. Leveraging large language models for nlg evaluation: A survey. *arXiv preprint arXiv:2401.07103*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Fina Polat, Ilaria Tiddi, and Paul Groth. 2024. Testing prompt engineering methods for knowledge extraction from text. *Semantic Web*. Under Review.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, pages 1–6.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan O Arik, and Tomas Pfister. 2023a. Better zero-shot reasoning with self-adaptive prompting. *arXiv preprint arXiv:2305.14106*.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Martin Eisenschlos, Sercan O Arik, and Tomas Pfister. 2023b. Universal self-adaptive prompting. *arXiv preprint arXiv:2305.14926*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

I2C-Huelva at SemEval-2024 Task 8: Boosting AI-Generated Text Detection with Multimodal Models and Optimized Ensembles

Alberto Rodero Peña, Jacinto Mata Vázquez, Victoria Pachón Álvarez
I2C Research Group, Universidad de Huelva

Abstract

With the rise of AI-based text generators, the need for effective detection mechanisms has become paramount. This paper presents new techniques for building robust models and optimizing training aspects for identifying synthetically produced texts across multiple generators and domains. The study, divided into binary and multilabel classification tasks, avoids overfitting through strategic training data limitation. A key innovation is the incorporation of multimodal models that blend numerical text features with conventional NLP approaches. The work also delves into optimizing ensemble model combinations via various voting methods, focusing on accuracy as the official metric. The optimized ensemble strategy demonstrates significant efficacy in both subtasks, highlighting the potential of multimodal and ensemble methods in enhancing the robustness of detection systems against emerging text generators. This strategy was applied to subtask A, monolingual classification, ranking 47th with an accuracy of 0.8079, and subtask B, multilabel classification, ranking 18th with an accuracy of 0.789.

1 Introduction

In the era of digital communication, AI-based text generators have become increasingly sophisticated, necessitating advanced detection methods to differentiate between human and machine-generated content (Radford et al., 2019; Brown et al., 2020). This paper addresses the challenge within the scope of English language texts, emphasizing the importance of reliable detection mechanisms in maintaining the integrity of digital discourse. The task at hand is crucial for various applications, including content moderation, misinformation prevention, and ensuring the authenticity of digital communication.

The core strategy of this system lies in its robustness and the optimization of model training. By limiting the size of the training dataset, the approach prevents models from overfitting to specific text generators, thereby enhancing their generalizability to novel content. Furthermore, the system leverages multimodal models that integrate traditional NLP techniques with numerical text features, such as lexical diversity and sentence structure, to enrich the detection capabilities. This is complemented by a rigorous exploration of ensemble methods and voting mechanisms to optimize model performance. Specifically, our conception of multimodal entails the strategic fusion of traditional, fine-tuned language models like RoBERTa with extracted numerical values from the text, such as number of grammatical errors and average sentence length, thereby enriching the language models with quantifiable text insights to enhance detection precision.

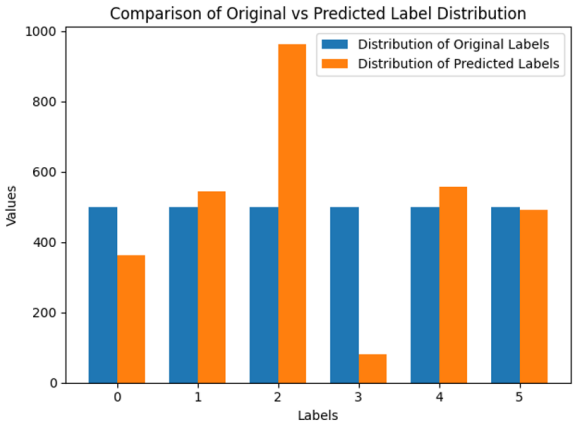


Figure 1: Comparison of Original vs Predicted Label Distribution Subtask B

Participation in this task led to the system ranking 47th in the monolingual subtask A with an accuracy of 0.8079 and 18th in the multilabel subtask B with an accuracy of 0.789 (Wang et al., 2024). These outcomes affirm the system's robustness in handling diverse generative models. However, the primary challenge encountered was distinguishing

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 845–852

between texts produced by similar generators. The system struggled to consistently differentiate between certain generators, often misattributing texts to one over another when faced with stylistically comparable outputs. This difficulty in discerning subtle variations between generator styles points to the need for further refinement in the detection algorithm, suggesting an area for future research to enhance the sensitivity and specificity of the model. As shown in Figure 1, the double bar graph compares the original label distribution and the predicted label distribution. The labels represent the model that generated each text of the samples, being 0 human, 1 chatGPT, 2 cohere, 3 davinci, 4 bloomz and 5 dolly.

2 Background

The task of detecting machine-generated text has garnered significant attention due to the rapid advancement and widespread use of AI-based text generators. The input for this task consists of textual content, with the output being a classification decision indicating whether the text is human or machine-generated in subtask A and which generator created the text in subtask B.

For this study, the dataset comprised English texts from diverse sources, including Wikipedia (March 2022 version), WikiHow, Reddit (ELI5), arXiv, and PeerRead (Koupaee and Wang, 2018; Kang et al., 2018). The machine-generated texts were produced using leading multilingual Large Language Models (LLMs) such as ChatGPT, textdavinci-003, LLaMa, FlanT5, Cohere, Dolly-v2, and BLOOMz. These models were prompted to create content resembling the human-written texts from the mentioned sources, ranging from Wikipedia articles to peer reviews and news briefs, ensuring a rich variety of genres and styles within the dataset. This richly varied dataset forms the foundation of the analysis, drawing on the comprehensive compilation of machine-generated texts as detailed in the work by Wang et al. (Wang et al., 2023). As shown in Figure 2, the training data distribution illustrates the sources and quantity of data used in the study.

This work focuses solely on the English portion of the dataset, engaging in the monolingual classification track. The choice of English allows for a concentrated examination of the nuances in detecting machine-generated texts in a language with extensive generative model research and development. The task setup and dataset composition are pivotal in understanding the challenges and innovations presented in this study.

This work builds upon foundational efforts in the field, such as "Machine-Generated Text Detection using Deep Learning" by Raghav Gaggar et al. (Gaggar et al., 2023), which emphasizes deep learning approaches for distinguishing AI-generated content. Gaggar's methodology leverages traditional neural network architectures, providing a critical basis for understanding how machine learning can be applied to text detection challenges. Similarly, "On the Possibilities of AI-Generated Text Detection" by Souradip Chakraborty et al. (Chakraborty et al., 2023) contributes to the discourse by establishing theoretical frameworks based on information theory, highlighting the nuanced differences between human and AI-generated texts and the implications for detection mechanisms. This paper underscores the importance of sample complexity and the robustness of detection systems to new and evolving text generators. "Ghostbuster: Detecting Text Ghostwritten by Large Language Models" by Vivek Verma et al. (Verma et al., 2023) methodology employs a series of weaker language models to compute token generation probabilities, offers a specialized perspective on model-agnostic detection. In contrast, this work extends the discourse by incorporating numerical text features alongside conventional NLP techniques within a multimodal framework, providing a more holistic analysis of text characteristics. This integration allows for a more nuanced distinction between human and AI-generated texts, addressing the challenges of style and generator diversity that single-model systems may struggle with.

3 System Overview

The system is designed to detect machine-generated text, combining an ensemble of finely-tuned transformer models such as RoBERTa ('FacebookAI/roberta-base') (Liu et al., 2019), ELECTRA ('google/electra-base-discriminator') (Clark et al., 2020), ALBERT (albert/albert-base-v2) (Lan et al., 2020), roberta-base-openai-detector ('roberta-base-openai-detector'), chatgpt-detector-roberta ('Hello-SimpleAI/chatgpt-detector-roberta') and BERT ('bert-base-uncased') (Devlin et al., 2018) with custom adaptations of RoBERTa including a one-vs-all system, that independently

Source/ Domain	Language	Total Human	Parallel Data						
			Human	Davinci003	ChatGPT	Cohere	Dolly-v2	BLOOMz	Total
Wikipedia	English	6,458,670	3,000	3,000	2,995	2,336	2,702	3,000	17,033
Reddit ELI5	English	558,669	3,000	3,000	3,000	3,000	3,000	3,000	18,000
WikiHow	English	31,102	3,000	3,000	3,000	3,000	3,000	3,000	18,000
PeerRead	English	5,798	5,798	2,344	2,344	2,344	2,344	2,344	17,518
arXiv abstract	English	2,219,423	3,000	3,000	3,000	3,000	3,000	3,000	18,000

Figure 2: Training Data Distribution

predicts each label's presence, treating each label as a separate binary classification problem, and multimodal models . A Random Forest classifier is used to analyze numerical text features. This ensemble integrates outputs from each model by aggregating predictions and confidence levels through various voting mechanisms. The process identifies the best combination of models and voting method, optimizing the ensemble to achieve the highest detection accuracy and robustness.

3.1 Training Sample Optimization for robustness

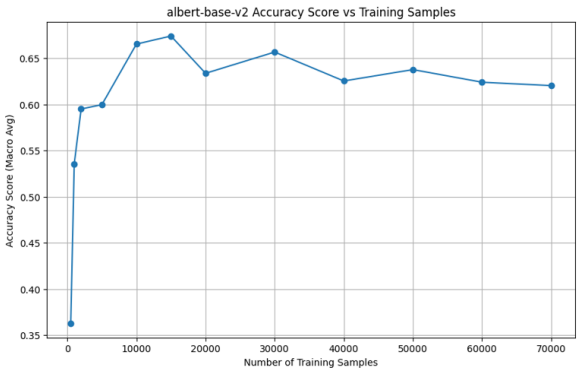


Figure 3: Accuracy by Training Sample Size

To optimize training samples for robustness, the number of samples used to train each model were systematically varied, aiming to find an optimal size that enhances robustness to new text generators while preventing overfitting. For instance, in the case of the ALBERT model, training began with 500 samples, then the model was reset and trained again with increasing sizes: 1000, 2000, 5000 samples, and so on. This process revealed that smaller sample sizes increased the model's robustness. It was determined that the ideal average number of samples for binary classification was 10,000, whereas multilabel classification required a larger average of 48,000 samples to maintain high predictive accuracy without compromising robustness to unseen generators in the evaluation dataset. As illustrated in Figure 3, the graph shows the relationship between model accuracy and training sample size.

3.2 Numerical Features

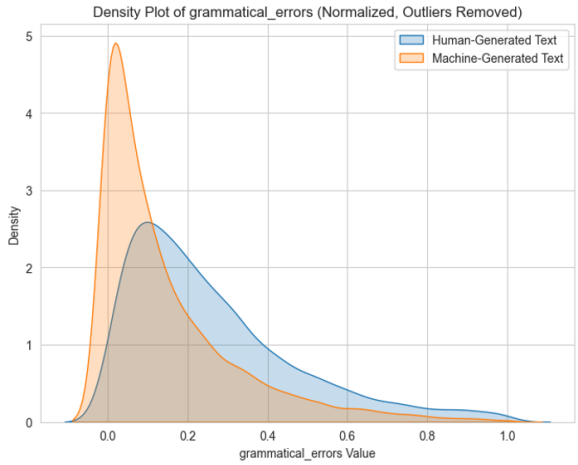


Figure 4: Human vs Machine Grammar Errors Density Plot

In addition to leveraging powerful transformer models, the system uniquely incorporates the extraction of numerical features from text to enhance its analytical depth. These features are: qord count, sentence count, lexical diversity, average sentence length, average word length, lexical density, flesch reading ease, gunning fog index, and grammatical errors in english, and they offer critical insights into the stylistic and structural elements of the text, which might be indicative of its origin. These features were obtained using methods from NLP libraries such as nltk. By analyzing these quantitative aspects, the system can identify subtle patterns and discrepancies that differentiate human-written texts from those generated by AI models, even when the linguistic content is convincingly human-like. This approach not only enriches the model's input but also helps in capturing the essence of text generation techniques used by various AI models, thereby contributing to a more robust detection

mechanism. The system's primary aim with numerical features was to supplement the multimodal model with additional information. For the numerical values, a Random Forest classifier was chosen as it is an easy out-of-the-box solution to be used with numerical values (Louppe, 2014). However, this aspect was not the main focus, and further experimentation was not pursued. Future work could explore the use of deep learning and other classification models like XGBoost to analyze these numerical features. As depicted in Figure 4, the density plot illustrates the distribution of grammar errors between human-written and machine-generated texts.

3.3 Multimodal Models

The system's architecture is notably enhanced by the inclusion of multimodal models, which not only utilize the capabilities of traditional NLP models like RoBERTa but also integrate numerical text features for a more comprehensive analysis. This approach, applicable to any large language model, involves extending the chosen LLM's architecture with a custom classification head that processes both the LLM's output and additional numerical features from the text. For this study, RoBERTa was selected due to its role in establishing the baseline performance, allowing for a direct comparison of the improvements attributed solely to the multimodal functionality. Two different multimodal models were used. The extended version includes all the numerical features extracted from the text, which performs better in binary classification but not as well in multimodal classification. The second model uses only the features that show a clear difference between texts written by humans and those generated by machines. This model does better in multilabel classification but doesn't do as well in binary classification. The numerical features used in the multimodal model are word count, average sentence length, average word length, gunning fog index and grammatical errors. The extended version also includes sentence count, lexical diversity, lexical density and flesch reading ease. It is also worth mentioning that the performance between multimodal versions is slight.

3.4 Optimization of Ensembles

The optimization of ensembles through various voting mechanisms stands as a testament to the system's strategic design. The system tested every combination of models to make sure each one added value to the ensemble and did not take away any useful information. Specifically, models are chosen for their complementary strengths and diverse natures, ensuring a broad coverage of the linguistic and stylistic features pertinent to text generation detection. It used the predictions and confidence scores from all included models along with the correct labels. Then, it applied different voting methods to see how they compared to the real labels. This way, it found the best mix of models and the best voting method. The voting methods tested included majority voting, majority score tie break voting (confidence based), rank voting, borda count voting and soft voting (Brownlee, 2020). In the binary classification task, a larger variety of models is employed to capture the nuanced differences between human and machine-generated texts, whereas for the multilabel task, only two models are needed, reflecting the different demands of each subtask. Notably, multimodal models, recognized for their high accuracy, are consistently selected across both subtasks, reinforcing the ensemble's performance. The chosen strategy ensures that the ensemble's collective judgment is both robust and sensitive to the nuances of text generation, significantly enhancing the system's overall accuracy and reliability.

As illustrated in Figure 5, the bar graph compares the accuracy of individual binary classification models, providing insights into their performance. The final ensemble model accuracy is also

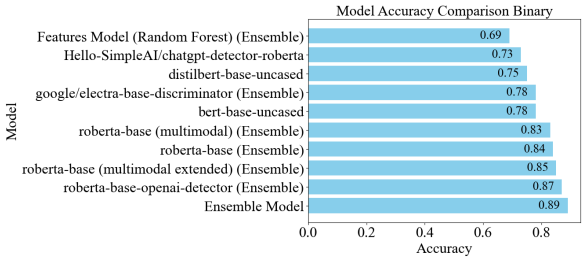


Figure 5: Substask A Models Accuracy in Training

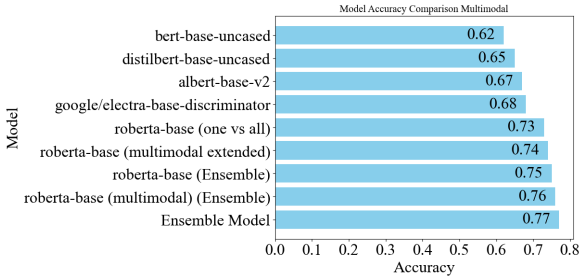


Figure 6: Subtask B Models Accuracy in Training

included. Similarly, Figure 6 presents a comparison of the accuracy of individual multilabel classification models. The figures illustrate the models that were ultimately chosen to be included in the ensemble using majority voting.

4 Experimental Setup

In this study, the optimization of the training sample size was a critical preliminary step before proceeding with the standard division of the dataset for model training and evaluation. The objective was to determine the most effective training sample size that would enable the models to learn sufficiently from the data without overfitting. This involved iterative testing of various sample sizes to identify the optimal balance that maximized model performance on unseen data. Once the ideal training sample size was established, it was then split following an 80-20 ratio, with 80% of the samples used for training and the remaining 20% for evaluation. The dev dataset served as the test set throughout the experiments, ensuring a consistent benchmark for evaluating the generalization ability of the models across different configurations and optimizations.

Fine-tuning the models was conducted with careful consideration of hyperparameters that directly influence model performance. The hyperparameters were determined by experimenting with a range of values and choosing those that led to better performance metrics. This approach aimed to enhance the model's ability to be robust to new text generators that were present in the evaluation dataset but not in the training dataset. The learning rate was set to 2e-5, a value chosen to ensure steady yet effective model updates without causing large fluctuations in model weights that could hinder learning. The batch size for both training and evaluation phases was maintained at 16, balancing computational efficiency with the need for granularity in gradient updates. The models underwent training for 3 epochs, a decision underpinned by the desire to minimize overfitting while allowing sufficient iterations for the models to converge to an optimal state. Weight decay was applied at 0.01 to regularize the model and further mitigate overfitting. The training process incorporated an epoch-based evaluation and save strategy, enabling continuous monitoring of model performance and retention of the best-performing model state at each epoch's conclusion, as determined by evaluation metrics.

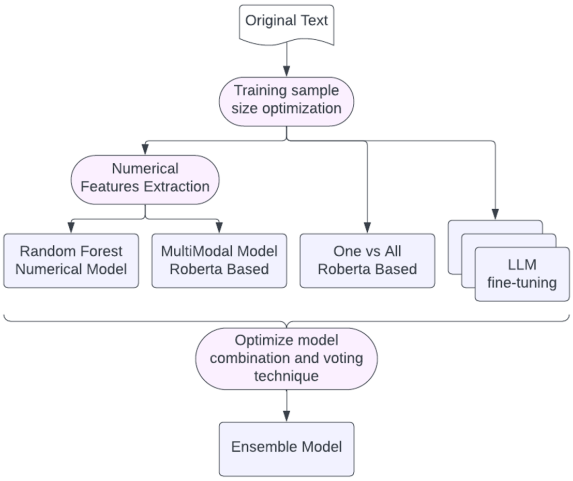


Figure 7: Experimental Setup for Training

The experimental framework utilized PyTorch for implementing the transformer-based models, specifically leveraging the RoBERTa-base model from the Hugging Face Transformers library for both the multimodal models and the one-vs-all classification approach in the multilabel subtask. For numerical feature extraction from text, the NLTK library was employed, enriching the model inputs with linguistic features that provide additional context and depth to the analysis. The numerical model, built using a Random Forest classifier, was optimized using the scikit-learn library, demonstrating the integration of traditional machine learning techniques with advanced NLP models for enhanced predictive performance. As depicted in Figure 7, the diagram illustrates the experimental setup for training, showcasing the steps and pipelines involved in the process.

5 Results

The system demonstrated commendable performance in the task, adhering to the official evaluation metric of accuracy. In Subtask A (monolingual classification), the system attained an accuracy of 0.8079, placing it at the 47th position in the competition. This ranking underscores the system's capability to effectively distinguish between human and machine-generated texts in a monolingual setting. For Subtask B (multilabel classification), the system achieved an accuracy of 0.789, ranking 18th out of the total number of participants. This notable performance highlights the system's robustness and effectiveness in handling more complex multilabel scenarios, despite the inherently

Models	Accuracy	F1	Precision	Recall	AUC
RoBERTa-base-openai-detector	0.87	0.86	0.94	0.78	0.87
RoBERTa-base	0.84	0.84	0.83	0.86	0.84
bert-base-uncased	0.78	0.78	0.78	0.79	0.78
google/electra-base-discriminator	0.78	0.78	0.77	0.79	0.78
distilbert-base-uncased	0.75	0.74	0.83	0.62	0.75
RoBERTa-base (baseline)	0.74	-	-	-	-
Hello-SimpleAI/chatgpt-detector-RoBERTa	0.73	0.72	0.91	0.52	0.73
Features Model (Random Forest)	0.69	0.69	0.71	0.64	0.69
Multimodal Models					
RoBERTa-base (multimodal extended)	0.85	0.85	0.86	0.83	0.85
RoBERTa-base (multimodal)	0.83	0.83	0.88	0.77	0.83
Ensemble model (submission)	0.89	0.89	0.87	0.91	0.89

Table 1: Results obtained for Subtask A

Models	Accuracy	F1	Precision	Recall
RoBERTa-base	0.75	0.72	0.73	0.75
RoBERTa-base (baseline)	0.75	-	-	-
RoBERTa-base (one vs all)	0.73	0.7	0.71	0.73
google/electra-base-discriminator	0.68	0.65	0.68	0.68
albert-base-v2	0.67	0.65	0.66	0.67
bert-base-uncased	0.62	0.62	0.62	0.66
distilbert-base-uncased	0.65	0.63	0.66	0.65
Multimodal Models				
RoBERTa-base (multimodal)	0.76	0.72	0.73	0.76
RoBERTa-base (multimodal extended)	0.74	0.71	0.73	0.74
Ensemble Model (submission)	0.77	0.73	0.73	0.77

Table 2: Results obtained for Subtask B

challenging nature of distinguishing between multiple generators. These metrics were obtained after applying the ensemble model for each task.

In a comprehensive evaluation using the evaluation dataset, tables comparing model performances shed light on the system's effectiveness. For Subtask A, comparisons between various models in binary classification, and specifically between the RoBERTa-base model and its multimodal extensions, reveal the somewhat superior performance of the multimodal models. These models, incorporating key numerical features, mostly outperformed other fine tuned classificators. A similar trend was observed in Subtask B's multilabel classification, where multimodal models again demonstrated some enhanced accuracy. This data, while not from the final test set, underscores the potential of multimodal approaches in effectively distinguishing between human and machine-generated texts across different classification scenarios.

Table 1 provides a comprehensive metrics comparison for binary classification including multimodal models. It highlights the performance of fine-tuned LLMs for binary classification, including a Features only model built with a Random Forest classifier, and the performance evolution from base model RoBERTa-base to advanced multimodal models that integrate numerical features. The ensemble model is also included in this table, showcasing its role in the collective modeling approach. Table 2 provides a similar comparison but for multilabel classification scenarios.

5.1 Quantitative Analysis

A series of studies and comparative analyses were conducted to dissect the impact of various design decisions, such as the optimization of training sample sizes, the integration of numerical features, and the selection of models within the ensemble. The dev dataset served as the primary test bed for these analyses, ensuring consistency in evaluating the system's modifications and optimizations.

- A notable finding was the system's increased performance when numerical features were integrated, suggesting the significant value these features add to understanding text beyond mere semantic analysis.

- The ensemble's optimized combination of models, including transformer-based and numerical models, was pivotal in enhancing accuracy. The binary classification required a more diverse set of models to capture the nuances of different text generators, whereas the multilabel task achieved high performance with just two, indicating the strategic importance of model selection based on the task's nature.

5.2 Error Analysis

The examination of errors, particularly for Subtask B, shed light on the complexity of multilabel classification. The system was tasked with identifying multiple generator labels within the same text, a challenge compounded by the nuanced differences between generators' styles. As depicted in Figure 8, the confusion matrix heatmap provides insights into the errors made by the system in multilabel classification.

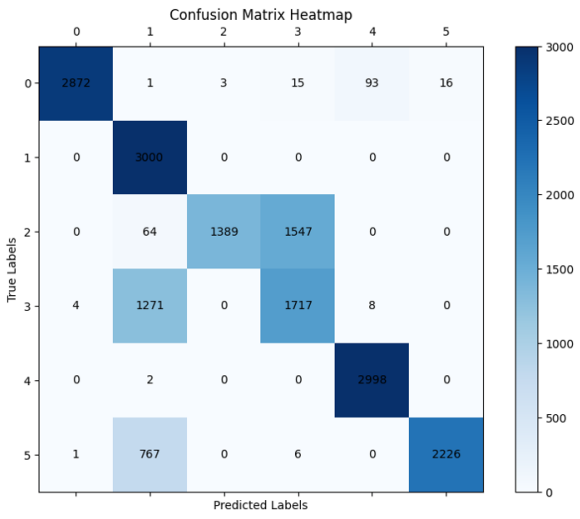


Figure 8: Confusion Matrix Heatmap Multilabel Subtask B

6 Conclusions

This study showcased innovative techniques aimed at enhancing model robustness in the task of detecting machine-generated text, notably through the careful optimization of training sample size, the strategic assembly of diverse models into optimized ensembles, and the deployment of multimodal models. These methodologies collectively facilitated a system that adeptly navigates the challenges of monolingual and multilabel classifications.

The exploration of training sample sizes revealed a delicate balance between sufficient model training and the avoidance of overfitting, highlighting the importance of dataset optimization. The ensemble model's success, derived from combining models with varying strengths, emphasizes the value of diversity in model architecture for robust performance. Moreover, the integration of multimodal models, blending traditional NLP techniques with numerical text features, showcased a sophisticated approach to capturing the nuanced distinctions between human and machine-generated texts.

Looking ahead, the focus will be on refining these novel techniques to further bolster model robustness. Future work will explore more granular adjustments to training sample sizes and investigate the potential of dynamic ensemble configurations responsive to the nature of the text being analyzed. Additionally, the extension of multimodal model frameworks to incorporate emerging linguistic and semantic features presents a promising avenue for enhancing detection capabilities. Applying these advanced methodologies to other areas of model building could advance the landscape of machine learning, offering a blueprint for developing systems that are not only robust but also universally applicable across various NLP tasks and challenges.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jason Brownlee. 2020. How to develop voting ensembles with python. https://machinelearningmastery.com/voting-ensembles-with-python/. Accessed: date.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023. On the possibilities of ai-generated text detection.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training

text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Raghav Gaggar, Ashish Bhagchandani, and Harsh Oza. 2023. Machine-generated text detection using deep learning.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4171–4186.

Gilles Louppe. 2014. Understanding random forests: From theory to practice. *arXiv*, 1407.7502.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.

Snarci at SemEval-2024 Task 4: Themis Model for Binary Classification of Memes

Luca Zedda and **Alessandra Perniciano** and **Andrea Loddo** and
Cecilia Di Ruberto and **Manuela Sanguinetti** and **Maurizio Atzori**
Department of Mathematics and Computer Science, University of Cagliari
Via Ospedale 72, Cagliari (Italy)
{luca.zedda, alessandra.pernician, andrea.loddo,
cecilia.dir, manuela.sanguinetti, atzori}@unica.it

Abstract

This paper introduces an approach developed for multimodal meme analysis, specifically targeting the identification of persuasion techniques embedded within memes. Our methodology integrates Large Language Models (LLMs) and contrastive learning image encoders to discern the presence of persuasive elements in memes across diverse platforms. By capitalizing on the contextual understanding facilitated by LLMs and the discriminative power of contrastive learning for image encoding, our framework provides a robust solution for detecting and classifying memes with persuasion techniques. The system was used in Task 4 of Semeval 2024, precisely for Substask 2b (binary classification of presence of persuasion techniques). It showed promising results overall, achieving a Macro-$F_1 = 0.7986$ on the English test data (i.e., the language the system was trained on) and Macro-$F_1 = 0.66777/0.47917/0.5554$, respectively, on the other three "surprise" languages proposed by the task organizers, i.e., Bulgarian, North Macedonian and Arabic. The paper provides an overview of the system, along with a discussion of the results obtained and its main limitations.

1 Introduction

In recent years, the natural language processing (NLP) community has witnessed an ever-growing number of contributions aimed at identifying and analyzing various forms of harmful language found on the Web, including offensive language (Zampieri et al., 2020), hate speech (Basile et al., 2019; Röttger et al., 2022)—also comprising misogyny and transphobia (Nozza et al., 2022; Kirk et al., 2023), and propaganda techniques (Da San Martino et al., 2019). These linguistic phenomena not only harm civil debate but can also fuel the polarization and radicalization of users' opinions.

In an increasingly multimodal context, particular attention has also been paid to memes (Dimitrov et al., 2021), which, due to their virality and communicative immediacy, can easily become key tools in online disinformation campaigns. Therefore, the development of techniques to effectively classify possible nuances of information manipulation within these forms of content sharing assumes a central role in online disinformation research.

This motivated our participation in the SemEval 2024 Task 4[1] (Dimitrov et al., 2024), which focuses on "Multilingual Detection of Persuasion Techniques in Memes". The task aims to develop models capable of identifying rhetorical and psychological techniques employed in memes to influence users' opinions. Our team participated in Substask 2b, consisting of a binary classification problem to determine whether a meme contains at least one persuasion technique among the predefined set of 22 techniques. The dataset released to participants is made up of memes with textual content in English. However, to assess the robustness of the systems during the evaluation phase, the organizers made test sets available in three other languages besides English, i.e., Arabic, Bulgarian, and North Macedonian.

For the purpose of this task, we developed a system that combines Large Language Models (LLMs) and contrastive learning image encoders to discern the presence of persuasive elements in memes across diverse platforms. The following sections will thus describe the system architecture and its deployment in the task. A discussion of the results obtained and of some most recurring errors will also be proposed, aiming to highlight possible research paths for the further improvement of the approach.

[1] https://propaganda.math.unipd.it/semeval2024task4/index.html

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 853–858

2 System Overview

The proposed method, named "Themis", is a modular neural network architecture designed to analyze multimodal data, specifically targeting memes that often contain both textual and visual elements.

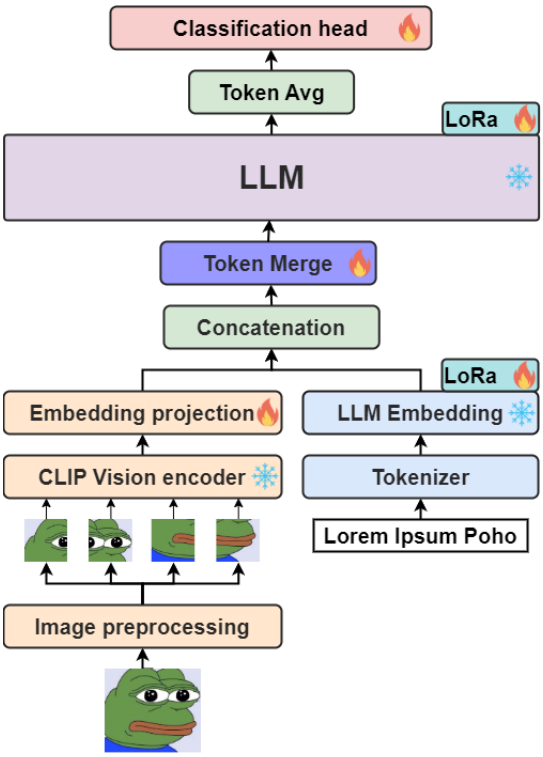


Figure 1: Themis model architecture.

2.1 Model Composition

Our Themis model comprises various interoperating components, as outlined in Figure 1. In this section, we shall examine their respective roles within the architecture.

Image Preprocessing. The meme image initially undergoes processing through the image processor of the Image Embedding Model, which standardizes and resizes it to conform to the model's specifications. Subsequently, the image is segmented into uniform patches of predetermined dimensions for subsequent processing. This preprocessing step guarantees that the image is adequately prepared for further analysis and feature extraction by the Image Embedding Model.

Image Embedding Model. Themis integrates an image embedding model for the extraction of features from meme images. This model is responsible for processing pixel values and extracting significant representations from the images.

Image Embedding Projection. An image embedding projection is applied to the features extracted from the image embedding model. This projection serves as a method to project image features in LLM-compatible size.

Large Language Model. Themis uses a language model to handle textual and visual inputs associated with memes. The LLM serves as the core of our model sequentially aligning tokens to an embedding space related to the persuasion detection task.

Token Merger Module. Themis employs a Token Merger module to merge tokens representing both image and textual features, enabling the model to attend to pertinent information within the images. This functionality allows the model to focus on salient aspects during meme processing. While drawing inspiration from the Patch Merger module (Renggli et al., 2022), our approach distinguishes itself by integrating both modalities. The Token merger learns a weight matrix that computes token scores based on representations and normalizes them using softmax. Subsequently, these weights are used to reduce the number of tokens through matrix multiplication. Ideally, this module aggregates similar tokens together, regardless of their original position. To address scale mismatches, layer normalization is applied post-merging, facilitating rapid adaptation through fine-tuning.

Token Average. We employ the token averaging technique, which involves extracting tokens from the LLM, to derive our final prediction. This strategy is designed to generate a single, semantically dense embedding, facilitating seamless processing by a classification head for obtaining the class prediction.

Classification Head. Themis incorporates a classification head to predict whether a meme contains specific persuasion techniques. This head takes the fused multimodal features and generates predictions based on the learned LLM representations.

2.2 Model Freezing and Low-Rank Adaptation (LoRA) Weights

The Themis model uses freezing techniques to control the training of certain parameters. Specifically, both the image embedding model and the language model are frozen during training. This ensures that

the pre-trained weights of these models are not updated, preserving the learned representations.

Additionally, Themis employs LoRA (Hu et al., 2022) weights to enhance its capabilities. LoRA weights are incorporated into the Image Embedding Projection layer and LLM model to introduce long-range interactions between tokens and patches, facilitating the capture of global context and improving overall performance in meme analysis tasks.

3 Experiment Setup

The experiments were executed on a workstation featuring an Intel Core i7-12700 @ 2.1GHz CPU, 32 GB RAM, and an NVIDIA RTX3060 GPU with 12GB of memory. Among the different experiments, one main issue is denoted by the limited availability of VRAM; this issue not only limited our approach to smaller LLM and Image encoders but also limited batch size. Our experiments aim to enable efficient prediction even in such low-end system requirements. As a result, we opted for a pre-trained Contrastive Language-Image Pre-Training (CLIP) (Radford et al., 2021) encoder as our main image encoder. Specifically, we used both CLIP Base[2] and CLIP Large[3] in our experiments. For the textual part, instead, we used TinyLlama[4] (Zhang et al., 2024), Phi-1.5[5] (Li et al., 2023) and Phi-2[6]. Table 1 depicts the full set of selected Image encoders and LLMs that suited our requirements.

Tipology	Model	# Params (B)
Image Encoder	CLIP Base 32	0.15
	CLIP Large 14	0.42
LLM	Phi-1.5	1.3
	Phi-2	2.7
	TinyLlama	1.1

Table 1: List of LLMs and Image Encoders used for our experiments.

For each combination of image and text models, the system was trained for 20 epochs, using a batch of 2 and a learning rate of $1e-4$, AdamW as the optimizer, and Binary Cross Entropy as the loss function.

To train the model we solely relied on the training data provided by the organizers. For Subtask 2b, this dataset comprised 1200 instances, each consisting of a meme image paired with its corresponding text. The validation set contained 150 instances, while the development set included 300 instances. The final test sets encompassed 600 memes in English, 100 memes each in Bulgarian and North Macedonian, and 160 memes in Arabic. The distribution of labels across the training, validation, and development sets is presented in Table 2[7].

Label	Train	Val	Dev
propagandistic	800	100	200
non-propagandistic	400	50	100

Table 2: Label distribution on the training, validation, and development set for Subtask 2b.

All results were evaluated using the official classification measure adopted by the task organizers for Subtask 2b, i.e., Macro-F_1.

To select our best model, we performed a search over the best set of hyperparameters. Specifically, we varied the rank of the LoRA weight matrices (LoRA R), their alpha regularization factor (LoRA Alpha), the dropout rate of the LoRA weights (LoRA Dropout), and most importantly, we controlled the number of tokens by a Token Merging strategy (see Section 2.1).

4 Results

The task was organized into two main evaluation phases: a development phase, during which only training and unlabeled development data were accessible, and a test phase, wherein the gold labels for the development set were disclosed alongside the unlabeled test sets in four languages: English, Arabic, Bulgarian, and North Macedonian. In this section, we outline the results obtained by our experiments in both phases.

Development phase. In this phase, we conducted tests on the unlabeled development set, employing various combinations of image encoders and LLMs. For each combination, we set LoRA R and LoRA

[2] https://huggingface.co/openai/clip-vit-base-patch32

[3] https://huggingface.co/openai/clip-vit-large-patch14

[4] In particular `TinyLlama-1.1B-Chat-v1.0`: https://huggingface.co/TinyLlama/TinyLlama-1.1B-Chat-v1.0

[5] https://huggingface.co/microsoft/phi-1_5

[6] https://huggingface.co/microsoft/phi-2

[7] For a comprehensive understanding of the dataset development and composition for each subtask, readers are encouraged to refer to the primary report of the task (Dimitrov et al., 2024).

Alpha to 8 and LoRA Dropout to 0.2. Notably, we omitted token merging during this phase based on preliminary results from the validation set, which indicated no significant performance enhancements with this setting. The results presented in Table 3 demonstrate that using larger Image Encoders, such as CLIP Large, yields an average increase of 0.7% in terms of Macro-F_1 performance. This enhancement may be attributed to higher-dimensional embeddings compared to their Base counterparts, even though it also produces a larger number of tokens due to a smaller patch size.

Image Encoder	LLM	Macro-F_1
CLIP Base	Phi-1.5	80.6
	Phi-2	80.6
	TinyLlama	80.8
CLIP Large	Phi-1.5	80.9
	Phi-2	81.6
	TinyLlama	81.6

Table 3: Macro-F_1 results on the development set of Subtask 2b across selected Image encoders and Large Language Models (for greater readability, F_1 scores are reported in percentage in all tables).

Among the various LLMs, both TinyLlama and Phi-2 exhibited identical performance. Consequently, we opted for TinyLlama and CLIP Large as the preferred models for further examination of model performance, using slightly adjusted hyperparameter settings. Specifically, we explored different numbers of tokens, LoRA ranks of 8, 16, and 32, and LoRA dropout values of 0.2, 0.3, and 0.4.

The results show that strong token merging strategies improve the model stability but limit its performance. The increase of the LoRA R greatly increases model instability due to the improved overfitting risk, while the increase in LoRA dropout greatly improves model performance, reaching the best Macro-F_1 result of 0.83487.[8] Our ablation study is depicted in Table 4.

Test phase. During the final evaluation phase of the campaign, we thus applied the best-performing setting described above on the test sets released by the task organizers. The results obtained are shown in Table 5. Overall, our team achieved reasonably good performance across both English and, albeit with a predictable decrease, in the zero-shot setting, where notable differences are observed. Upon comparing our performance with each top-ranked system in this subtask, we observe that the absolute difference between our system and the best-performing system in English (i.e., LMEME, which also ranks as the top system for Bulgarian) is 0.012, indicating that Themis achieved results very close to the top performer. For Bulgarian, the absolute difference is even smaller, at 0.003, suggesting that both systems exhibit very similar performance in this language. Conversely, for Arabic and North Macedonian, the difference is more pronounced, at 0.059 and 0.207, respectively, underscoring the limitations of our system in these languages.

[8] See the task leaderboard at https://propaganda.math.unipd.it/semeval2024task4/SemEval2024task4_dev.html

# tokens	LoRA R	LoRA Dropout	Macro-F_1
-	8	0.2	81.6
-	16	0.2	81.7
-	32	0.2	79.1
64	8	0.2	81.7
96	8	0.2	80.0
128	8	0.2	78.8
192	8	0.2	77.1
-	8	0.3	82.8
-	8	0.4	**83.4**

Table 4: Ablation study. In-depth results over the development set using CLIP Large and TinyLlama and different combinations of LoRA ranks (LoRA R column) and dropouts (LoRA Dropout column).

Language	Rank	Macro-F_1	Micro-F_1
English	5	79.8	82.6
Bulgarian	2	66.7	84.0
North Macedonian	8	47.9	72.0
Arabic	7	55.5	55.6

Table 5: Official results obtained across different languages on the test set.

5 Discussion and Error Analysis

Despite achieving promising results, in terms of Macro-F_1 scores, our model still occasionally misclassifies instances, particularly in cases involving the propagandistic nature of memes. Although the labeled test set was not made available by the organizers, we were still able to inspect more in detail the results obtained on the development set. Figures 3 and 4 illustrate examples of false positive and false negative predictions, respectively. In both

instances, we can formulate hypotheses regarding why and when our model generates errors. The false positive example may be misconstrued as employing a "Slogan" based persuasion technique, possibly due to text present in the sign. The false negative could stem from the model's inability to recognize inherent sarcasm due to limited sarcastic examples, further train with a larger dataset could mitigate this issue. The confusion matrix depicted in Figure 2 reveals an uneven distribution of errors across both classes, indicating a bias towards the propagandistic class. This bias could be attributed to the imbalance in the training set.

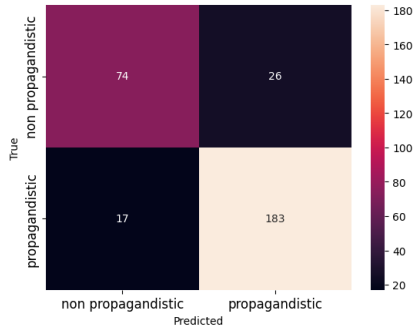


Figure 2: Confusion matrix of Themis predictions on the development set.

Figure 3: Example of false positive.

6 Conclusions

In this study, we introduced Themis, a novel model for analyzing multimodal memes by integrating LLMs and contrastive learning image encoders. Through comprehensive experiments, Themis demonstrated remarkable efficacy in detecting persuasion techniques within memes, achieving a notable F1 score of up to 83.4%. Our findings

Figure 4: Example of false negative.

underscore the critical role of meticulous model architecture design and hyperparameter optimization in meme analysis tasks. Notably, Themis presents a robust solution to combat societal challenges posed by biased content online, offering a promising avenue for mitigating the spread of misinformation and promoting digital discourse integrity.

Code availability

The code for our Themis model and train strategy is available on GitHub at: https://github.com/demonprin/Themis-SEMEVAL-public

Acknowledgements

The work has been partially supported by the project DEMON "Detect and Evaluate Manipulation of ONline information" funded by MIUR under the PRIN 2022 grant 2022BAXSPY (CUP F53D23004270006, NextGenerationEU), by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU (NextGenerationEU), by project eINS Ecosystem of Innovation for Next Generation Sardinia (CUP F53C22000430001) under the NRRP MUR program funded by the EU - NGEU (NextGenerationEU) and project NEST "Network 4 Energy Sustainable Transition–NEST" (CUP F53C22000770007) under the NRRP MUR program funded by the EU - NGEU (NextGenerationEU).

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Detecting propaganda techniques in memes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6603–6617, Online. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hannah Kirk, Wenjie Yin, Bertie Vidgen, and Paul Röttger. 2023. SemEval-2023 task 10: Explainable detection of online sexism. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2193–2210, Toronto, Canada. Association for Computational Linguistics.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*.

Debora Nozza, Federico Bianchi, Anne Lauscher, and Dirk Hovy. 2022. Measuring harmful sentence completion in language models for LGBTQIA+ individuals. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 26–34, Dublin, Ireland. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR. ISSN: 2640-3498.

Cédric Renggli, André Susano Pinto, Neil Houlsby, Basil Mustafa, Joan Puigcerver, and Carlos Riquelme. 2022. Learning to merge tokens in vision transformers. *ArXiv*, abs/2202.12015.

Paul Röttger, Haitham Seelawi, Debora Nozza, Zeerak Talat, and Bertie Vidgen. 2022. Multilingual HateCheck: Functional tests for multilingual hate speech detection models. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 154–169, Seattle, Washington (Hybrid). Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model.

Fired_from_NLP at SemEval-2024 Task 1: Towards Developing Semantic Textual Relatedness Predictor - A Transformer-based Approach

Anik Mahmud Shanto, Md. Sajid Alam Chowdhury, Mostak Mahmud Chowdhury, Udoy Das, Hasan Murad
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
u1904{049, 064, 055}@student.cuet.ac.bd, u1804109@student.cuet.ac.bd,
hasanmurad@cuet.ac.bd

Abstract

Predicting semantic textual relatedness (STR) is one of the most challenging tasks in the field of natural language processing. Semantic relatedness prediction has real-life practical applications while developing search engines and modern text generation systems. A shared task on semantic textual relatedness has been organized by SemEval 2024, where the organizer has proposed a dataset on semantic textual relatedness in the English language under Shared Task 1 (Track A3). In this work, we have developed models to predict semantic textual relatedness between pairs of English sentences by training and evaluating various transformer-based model architectures, deep learning, and machine learning methods using the shared dataset. Moreover, we have utilized existing semantic textual relatedness datasets such as the stsb multilingual benchmark dataset, the SemEval 2014 Task 1 dataset, and the SemEval 2015 Task 2 dataset. Our findings show that in the SemEval 2024 Shared Task 1 (Track A3), the fine-tuned-STS-BERT model performed the best, scoring 0.8103 on the test set and placing 25th out of all participants.

1 Introduction

Nowadays, there have been notable advancements in understanding and measuring pairwise semantic relatedness between texts within the domain of natural language processing. Predicting semantic relatedness plays a significant role in improving search engines, question-answering systems, text summarization tools, and machine translation.

However, previous works in natural language processing have mainly dealt with semantic similarity, a smaller aspect of relatedness, mainly due to the limited availability of relatedness datasets. Besides, dealing with ambiguous words or phrases that have multiple meanings can make semantic relatedness difficult. Understanding cultural context in language has been complex, and existing models have struggled to capture these variations. As language evolves, models struggle to adapt quickly to new linguistic patterns and expressions. To bridge these gaps, we need improved models that understand not just words but also context, cultural differences, and how language changes over time.

Semantic relatedness models have been developed using various transformer-based, deep learning, and machine learning techniques. Traditional machine learning methods (Buscaldi et al., 2015) have relied on predefined rules and features and offered moderate results. These approaches have often struggled with complex semantic relationships. Deep learning-based (Wang et al., 2018) approaches have surpassed traditional machine learning models in capturing complex relationships, particularly in tasks requiring a deep understanding of context. However, transformer-based approaches (Devlin et al., 2019) have outperformed others when it comes to capturing semantic relationships, particularly in understanding context, managing long-range dependencies, and handling contextual embeddings.

SemEval has arranged a shared task named SemEval 2024 Task 1: Semantic Textual Relatedness (STR) (Ousidhoum et al., 2024b), introducing a novel dataset called Shared Task 1 (Track A3) (Ousidhoum et al., 2024a) for determining the level of pairwise semantic relatedness between sentences based on the similarity score that ranges from 0.0 to 1.0.

The primary goal of this task is to build a robust and accurate model to predict the semantic relatedness between pairs of English sentences.

To accomplish this goal, we have used a variety of models, incorporating machine learning models (Linear Regression, Random Forest, XGBoost), models of deep learning (LSTM, BiLSTM), and pre-trained models based on transformer (RoBERTa, bert-base-uncased). We have named our approach of using the bert-base-uncased

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 859–864

model as STS-BERT.

By training and assessing every model, we have carried out a comparison analysis on the Semeval 2024 Task 1 (Track A3) dataset (Ousidhoum et al., 2024a), STSB multilingual dataset (May, 2021), SemEval 2014 Task 2 dataset (Marelli et al., 2014) and dataset provided for Task 1 in SemEval 2015 (Agirre et al., 2015) and have finally come to a conclusion that the STS-BERT model has demonstrated better performance compared to others boasting an impressive Spearman correlation coefficient of 0.81033 on the test dataset.

Key contributions of our research work are listed below -

- We have developed a fine-tuned-STS-BERT model that significantly helps in accurately predicting semantic textual relatedness across diverse sentence pairs.
- We have evaluated the model's performance through various tests conducted using the dataset and subsequently performed an in-depth evaluation of the outcomes.

The GitHub repository that follows has the implementation details available - https://github.com/Fired-from-NLP/SemEval-2024-task-1-track-A-eng.

2 Related Works

The associated works on semantic textual similarity can be generally categorized into three parts, approaches focused on machine learning, deep learning, and attention-based mechanism (transformer).

Among machine learning models, the Support Vector Regression model has been applied for calculating the semantic relationship between two short sentences (Sultan et al., 2013). In this system, three distinct measures, namely overlap in word n-gram, overlap in character n-gram, and semantic overlap, have been used for predicting similarity. In (Buscaldi et al., 2015), a Random forest-based approach has been utilized to find the semantic sentence similarity. The approach has relied on various similarity measures such as WordNet-based conceptual similarity, IC-based similarity, syntactic dependencies, and information retrieval-based similarity.

Traditional deep learning methods have depended on single or multiple granularity representations for detecting similarity. Apart from that, a different architecture that has focused on multiple positional sentence representations has been proposed (Wang et al., 2018). It has used Bi-LSTM for generating representations that enable the model to capture better context understanding. Another architecture has introduced a Siamese adaptation of LSTM (Mueller and Thyagarajan, 2016). Using a fixed-sized vector and a simple Manhattan metric, the model transforms sentence representation that represents semantic relationships. Another paper has described an architecture that has been built using deep learning paradigms (Zhao et al., 2015). This architecture has been trained using a combination of features like features based on a string, features based on a corpus, and features based on syntactic similarity, as well as newer matrices derived from distributed word embedding.

Transformer-based approaches have surpassed both machine learning and deep learning models in calculating semantic sentence relationships. Unlabeled text can be used to pre-train deep bidirectional representations using BERT (Devlin et al., 2019). BERT can be fine-tuned to do various NLP-related tasks like semantic analysis. A replication of BERT called RoBERTa (Liu et al., 2019), has focused on hyperparameters and training data size to improve model performance.

In this shared task, we have used BERT-based pre-trained models as they have been proven to be superior to other models available.

3 Dataset

We have employed the dataset made available as part of Shared Task 1 (Track A3) of the SemEval 2024: Semantic Textual Relatedness (STR) which contains 5500 samples in the training dataset and 250 samples in the dev dataset. Besides, the stsb multilingual benchmark dataset (May, 2021), the SemEval 2014 Task 1 dataset (Marelli et al., 2014) and the Semeval 2015 Task 2 dataset (Agirre et al., 2015) have been used.

Task	Sentence Pairs		
	Train	Validation	Test
SemEval 2014	4500	500	4928
SemEval 2015	2997	750	6729
stsb-multi-mt	5749	1500	1379

Table 1: Data sizes for external datasets

Table 1 shows the distribution of samples that we have used from external datasets. These datasets have been merged to get a total of 32508 samples

and then divided further into two sets: train and validation comprising 25676 and 6732 samples respectively. We have replaced the similarity score of duplicate sentence pairs with the average value to avoid labeling biases among different datasets. For the test dataset, The dataset made available as part of Shared Task 1 (Track A3), which consists of 2600 samples, has been used. These datasets

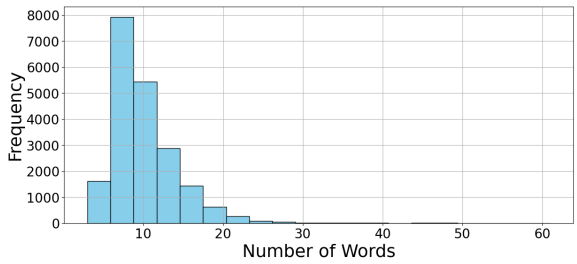


Figure 1: Word distribution of sentence1

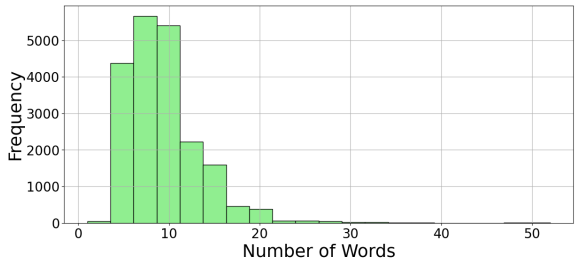


Figure 2: Word distribution of sentence2

contain a pair of sentences in each row, which we have split into two separate sentences namely sentence1 and sentence2. Figure 1 and Figure 2 show that sentence1 contains an average of 6-12 words, while sentence2 contains 3-12 words.

4 System Overview

In this section, we have outlined our methodology to develop models for determining sentence relatedness. First, we have used various extraction strategies to extract characteristics and then utilized a variety of machine learning and deep learning algorithms. Moreover, we have employed different transformer models to develop the system. Figure 3 provides a summary of our working methods.

4.1 Machine Learning-based Approaches

For determining sentence relatedness, we have applied traditional Machine learning-based methods such as Linear Regression and Random Forest. Moreover, To increase the performance, we have employed an ensemble classifier called XGBoost.

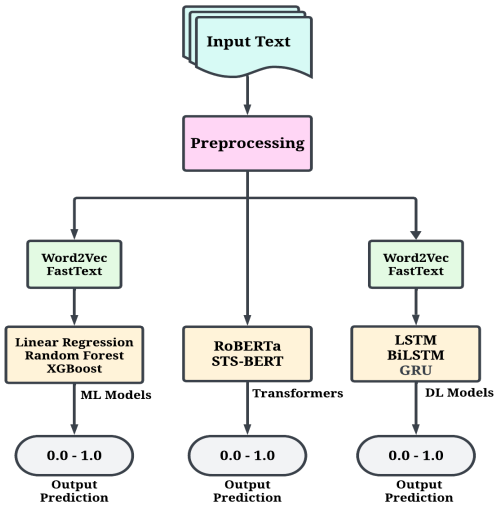


Figure 3: An outline of our approach

Here, we have tokenized the dataset using NLTKTokenizer, and then have we used Word2Vec to extract features. We also have used FastText for feature extraction as it not only captures semantic meaning like Word2Vec but also encodes subword information, allowing it to handle out-of-vocabulary words and morphologically rich languages more effectively. We have set the number of decision trees or boosting rounds to n_estimators for the ensemble approach at 100.

4.2 Deep Learning-based Approaches

Deep learning-based models have been utilized for determining sentence relatedness. We have implemented both models based on LSTM and Bi-LSTM. Two LSTM layers with various numbers of LSTM cells have been applied to the LSTM model. Each of the two directional layers has 50 or 100 LSTM cells in it. We have employed two Bi-LSTM layers, each with 100 and 50 Bi-LSTM cells, in the Bi-LSTM model.

4.3 Transformer-based Approaches

Methods based on transformers are now widely employed in many different contexts. We have employed STS-BERT (Devlin et al., 2019) and RoBERTa to tackle this task. As the sentences can be diverse, having a single representation and better understanding of the sentences is very important. For this reason, we have used the feature vector of the pooling layer as shown in Figure 4.

In our approach, we have first split the pair of sentences in the dataset into two. We have used two bert-based-uncased (Devlin et al., 2019) for

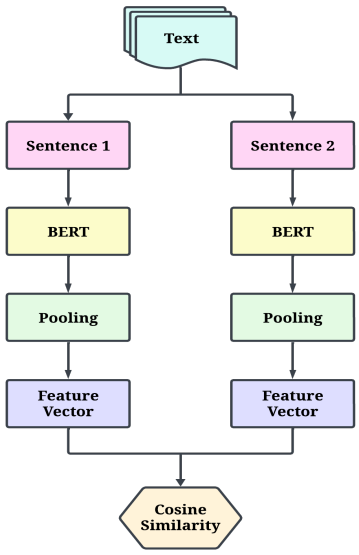


Figure 4: STS-BERT: Transformer-based model architecture for predicting semantic textual relatedness

two sentences. We have obtained the feature vector from the pooling layer of these two Bert models. After obtaining the pooled embeddings, we feed them to the cosine similarity for performing the relatedness task and compare them with the ground truth relatedness score. Then, we compute the loss using MSE (Mean Squared Error) based on the predicted relatedness and the actual relatedness. After the loss is calculated to improve the performance and minimize the loss, we have updated the model parameters using gradient descent.

5 Experimental Setup

This section gives a summary of our experimental setup while training and evaluating our model architectures for semantic textual relatedness.

5.1 Environment Setting

The simulation was executed on a personal computer featuring an Intel Core i7-9700 CPU clocked at 3.00 GHz and an NVIDIA GeForce GTX 2060 GPU. Additionally, to ensure ample processing capability, a Kaggle Notebook equipped with a P100 GPU was utilized.

5.2 Data Preparation

Besides the dataset provided in this competition, we have used three external datasets. We have used the stsb multilingual benchmark dataset (May, 2021), the SemEval-2014 Task 1 dataset (Marelli et al., 2014), and the SemEval-2015 Task 2 dataset (Agirre et al., 2015). We have combined all three datasets. The similarity score of external datasets ranges from 0.0 to 5.0. However, the provided dataset for this competition holds the relatedness between sentences ranging from 0.0 to 1.0. We multiplied the relatedness score of the dataset offered in the competition by 5.0 to match the similarity score in the combined dataset. We have replaced the similarity score of duplicate sentence pairs with the average value. Then, we have split the combined dataset into the training dataset and the validation dataset. The final size of the training dataset is 25676, whereas the overall size of the validation dataset is 6732. We have used the test dataset provided in the competition. The test set contains 2600 samples.

5.3 Parameter Settings

Table 2 shows the parameter settings used in LSTM. BiLSTM, and RoBERTa models.

Model	lr	optim	bs	epoch
LSTM	$1e^{-6}$	Adam	32	10
BiLSTM	$1e^{-6}$	Adam	32	10
RoBERTa	$1e^{-6}$	Adam	32	12

Table 2: Parameter configurations for various models

In Table 2, learning rate, optimizer, batch size, and number of epochs are represented by the variables lr, optim, bs, and epoch, in that order.

Table 3 summarizes the parameter settings used in our proposed STS-BERT model.

Parameter	Value
Learning Rate	1×10^{-6}
Optimizer	AdamW
Batch Size	8
Number of Epochs	12
Loss Function	Mean Squared Error (MSE)
Pooling	Mean Pooling

Table 3: Model parameter settings.

5.4 Evaluation Metrics

The instruction of Shared Task 1 of SemEval 2024 has been to use the Spearman correlation to evaluate the performance of our model using the test dataset. The mathematical representation of the Spearman correlation is provided in equation 1. Besides, we have used Cosine similarity in our model

to predict similarity between sentences. Equation 1 presents the mathematical representation for Cosime similarity.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (1)$$

In this context, ρ denotes the Spearman correlation coefficient. d_i represents the difference between the ranks of corresponding observations in the two variables, while n indicates the total number of observations.

$$\text{cos\_sim}(A, B) = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

Where, $\text{cos\_sim}(A, B)$ is the cosine similarity between vectors A and B. A_i and B_i denote the components of vectors of A and B respectively. n indicates the dimensionality of the vectors.

6 Experimental Results

In this section, we have showcased the experimental findings obtained during the training and evaluation stages of the proposed model for semantic textual relatedness prediction.

Table 4 presents a comparative analysis of different types of models, evaluating their performance using the Spearman correlation coefficient on the test dataset.

Category	Model	Embedding	Score
ML	Linear Regression	word2vec	0.0507
		fasttext	0.0507
	Random Forest	word2vec	0.1298
		fasttext	0.1198
	XGBoost	word2vec	0.3178
		fasttext	0.2072
DL	LSTM	word2vec	0.445
		fasttext	0.420
	BiLSTM	word2vec	0.4990
		fasttext	0.429
BERT	RoBERTa	-	0.749
	STS-BERT	-	**0.810**

Table 4: Results of different models on the test dataset

Among the machine learning models, we have found that the XGBoost model with word2vec embedding has achieved the highest score of 0.3178. In the deep learning category, we have seen better performance as both LSTM and BiLSTM models have higher scores than the machine learning models. The BiLSTM model achieved a score of 0.499, slightly outperforming the LSTM model, which obtained a score of 0.445.

In some cases, Fasttext word embedding has obtained the best results compared to word2vec (Meden, 2022). Therefore, we have also tested the performance of the model using Fasttext embedding. However, the transformer-based models have clearly outperformed other models based on machine learning and deep learning. For instance, the RoBERTa model achieved a score of 0.749, while our proposed STS-BERT model demonstrated exceptional performance with an impressive score of 0.810.

7 Error Analysis

In the development phase external datasets, SemEval 2014 Task 1 (Marelli et al., 2014), SemEval 2015 Task 2 (Agirre et al., 2015) and multilingual benchmark dataset (May, 2021) along with the competition dataset have been utilized. Hence the training set becomes more diverse and our model fails to learn about the relatedness between the sentences. The similarity scores of the external datasets ranged between 0.0 to 5.0. To make all the scores similar we have multiplied the scores of the competition dataset by 5.0 and normalized the whole training set by dividing all the scores by 5.0. Due to multiple conversions of the range of scores, precision loss has occurred. Sentence transformation has been another key reason for the poor performance of the model. When the second sentence is the transformation of the first sentence, our model can not detect it. For example, if the first sentence is in simple form and the second sentence is in the complex form of the first sentence, the model shows poor performance in that case. As a result, the overall performance of our proposed system has degraded.

8 Conclusion

In this research, we have conducted a comparative performance analysis, assessing a range of machine learning, deep learning, and transformer-based models to predict the semantic textual relatedness between pairs of English sentences. We have utilized the Task 1 (Track A3) dataset provided in the shared task, along with additional external datasets, for training various models. Our results indicate that the STS-BERT model has outper-

formed all other models, achieving an impressive score of 0.810. However, after analyzing errors, we have discovered that the slight score decrease is due to the integration of large external STS datasets with varying output ranges. To address this in future work, we plan to implement alternative strategies. Moreover, we will work on Task 1 (Track B and C) to have more comprehensive findings.

9 Ethical Considerations

To advance semantic text relatedness, we commit to emphasizing privacy through informed consent, reducing biases, as well as transparent modeling. Our ethical position prioritizes responsibility, accessibility, and privacy to build a positive and open technology environment.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *SemEval 2015*, Denver, Colorado. ACL.

Davide Buscaldi, Jorge García Flores, Ivan V. Meza, and Isaac Rodríguez. 2015. SOPA: Random forests regression for the semantic textual similarity task. In *SemEval 2015*, Denver, Colorado. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval 2014*, Dublin, Ireland. ACL.

Philip May. 2021. Machine translated multilingual sts benchmark dataset.

Katja Meden. 2022. Semantic Similarity of Parliamentary Speech using BERT Language Models & fastText Word Embeddings.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Md. Sultan, Steven Bethard, and Tamara Sumner. 2013. DLS@CU-CORE: A simple machine learning model of semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. ACL.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2018. A deep architecture for semantic matching with multiple positional sentence representations. In *ACL*.

Jiang Zhao, Man Lan, and Jun Feng Tian. 2015. ECNU: Using traditional similarity measurements and word embedding for semantic textual similarity estimation. In *SemEval 2015*. ACL.

BITS Pilani at SemEval-2024 Task 1: Using text-embedding-3-large and LaBSE embeddings for Semantic Textual Relatedness

Dilip Venkatesh[1] and **Sundaresan Raman**[1]

[1]Birla Institute of Technology and Science, Pilani, Rajasthan, India
{f20201203, sundaresan.raman}@pilani.bits-pilani.ac.in

Abstract

Semantic Relatedness of a pair of text (sentences or words) is the degree to which their meanings are close. The Track A of the Semantic Textual Relatedness shared task aims to find the semantic relatedness for the English language along with multiple other low resource languages with the use of pretrained language models. We propose a system to find the Spearman coefficient of a textual pair using pretrained embedding models like **text-embedding-3-large** and **LaBSE**.

1 Introduction

Semantic relatedness is defined as the degree of closeness of textual units (sentences, words, paragraphs) (Mohammad, 2008, Mohammad and Hirst, 2012). This makes semantic relatedness an important metric to understand the meaning of text. A paragraph is a string of multiple related sentences and similar paragraphs in a sequential manner form passages or documents which provides valuable information. Understanding this cohesion among sentences (Bernhardt, 1980) and passages is critical for understanding meaning and therefore generating more powerful natural language processing systems. We consider text to be semantically close if there is some sort of similar meaning. We can see an example of textual relatedness in Table 1

We make an important differentiation between **Semantic relatedness** and **Semantic Similarity**. Semantic similarity is when two textual units are synonymous, hyponymous, antonymous, or troponymous relation between them (Abdalla et al., 2023). Semantic relatedness consists of when there is a lexical relation between two units of text *conductor-orchestra, teacher-book*.

Since semantic relatedness is crucial to understanding meaning, it has many use cases in various NLP tasks such as question answering and text generation to produce coherent statements (Abdalla et al., 2023). Other natural language challenges like machine translation or information retrieval can be reduced to a semantic distance problem. It is also a key factor for text summarization, the relation between sentences in text will allow for more accurate summaries without too much loss of context.

For Track A of the SemEval 2024 Task 1: *Semantic Textual Relatedness **(STR)*** (Ousidhoum et al., 2024b) on Codalab (Pavao et al., 2023), we aim to create a system to automatically detect the degree of semantic relatedness between pairs of sentences with the OpenAI *text-embedding-3-large* and LaBSE text embedding models. This is for languages like English, as well as multiple low resource languages like *Algerian Arabic, Amharic, Hausa, Kinyarwanda, Marathi, Moroccan Arabic, Spanish, Telugu*.

Our code can be found on GitHub at `https://github.com/dipsivenkatesh/SemEval-2024-Task-1`

2 Background

2.1 Task and Data Description

The Semantic Textual Relatedness shared task [1] consists of three tracks.

- **Track A**: Supervised
- **Track B**: Unsupervised
- **Track C**: Cross-lingual

In this paper we go through our team's system to solve the track A of the challenge.

For the first track, we must develop a system to automatically find the closeness of meanings (semantic relatedness) between two sentences. We need to generate a relatedness score between 0

[1]`https://codalab.lisn.upsaclay.fr/competitions/16799`

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 865–868
June 20-21, 2024

PAIRS	SENTENCE 1	SENTENCE 2
1	There was a lemon tree next to the house.	The boy enjoyed reading under the lemon tree.
2	There was a lemon tree next to the house.	The boy was an excellent football player.

Table 1: Sentence relatedness: We can see that the sentences in pair 1 are more related than the sentences in pair 2

(completely unrelated) and 1 (maximum relations). For this track teams are allowed to submit systems that use the given datasets or any external datasets. The use of pre-trained language models are also allowed.

The datasets for training consisted of a pair of sentences and the 0 to 1 semantic relatedness scores graded through manual annotation. A comparative annotation approach was used for generating these gold label scores thereby avoiding biases of traditional rating and guaranteeing a high reliability.

2.2 Previous Work

In recent times the standard way to represent word meanings is as **vector semantics**. This comes from two major ideas, the idea to represent a word in three dimensional vector space (Osgood et al., 1957) and defining a word by the distribution of words around it (Harris, 1954 and Joos, 1950). Representing text as embeddings is an example of representation learning (Bengio et al., 2013).

The combination of term frequency (Luhn, 1957) and inverse document (Sparck Jones, 1972) frequency led to the use of **tf-idf** for representing word embeddings. Tf-idf had many faults, it did not represent contextual word relationships or word co-occurrence. This is fixed with in **Pointwise Mutual Information** (PMI) (Fano and Wintringham, 1961) a measure of how frequent two events occur, compared their occurrence if they were independent. The problem with tf-idf and PMI embeddings is that they are sparse vectors. Instead methods like **word2vec** (Mikolov et al., 2013) and **GloVe** (Pennington et al., 2014) produce dense vectors for word embeddings. Language models have gained a lot of traction due to their understanding of natural language. Language models like BERT have the ability to generate contextual embeddings. Contextual embeddings are used to represent the word in the context that it is used.

More recently, state of the art embedding models use pre-trained transformers by fine tuning the to a certain task. This is used in text embdding models like **Sentence-BERT** (SBERT) (Reimers and Gurevych, 2019) that uses BERT along with siamese and triplet network structures to generate sentence embeddings.

For the Semantic Textual Relatedness shared task track A our system, uses OpenAI's text-embedding-3-large to generate text embeddings. We also use the Language-agnostic BERT Sentence Embedding model (LaBSE) (Feng et al., 2022) to generate embeddings for the other languages as this model generates better representations for these languages.

2.3 Evaluation Metrics

There are multiple ways to evaluate relatedness using the vector embeddings of the text, dot product is one such metric. However it favors longer vectors, therefore normalized dot product or the cosine of the angle between the two vectors is used

The evaluation metric for this challenge was the Spearman rank coefficient which compares the the sentence relatedness predictions of the system against the gold truth human judgements. The Spearman rank coefficient (ρ) can be calculated as:

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}$$

where:

- d_i is the difference between the ranks of corresponding variables x_i and y_i,
- n is the number of observations.

3 System Overview

3.1 text-embedding-3-large

We use OpenAI's latest large text embedding model **text-embedding-3-large**[2] to generate embeddings. The state of the art *text-embedding-3-large* creates embeddings of 3072 dimensions. The embedding model achieves a score of 54.9% MIRACL benchmark (Zhang et al., 2023) and 64.6% on the MTEB benchmark (Muennighoff et al., 2022).

[2] https://openai.com/blog/new-embedding-models-and-api-updates

Model / Language	amh	arq	ary	eng	esp	hau	kin	mar	tel
LaBSE	0.79	0.46	0.41	N/A	0.72	0.48	0.50	0.80	0.78
text-embedding-3-large	0.68	0.56	0.45	0.86	0.70	0.47	0.52	0.78	0.74

Table 2: Performance comparison of sentence-transformers/LaBSE and text-embedding-3-large on training set

3.2 LaBSE

We use the **Language-agnostic BERT Sentence Embedding** (LaBSE) model (Feng et al., 2022) to generate the embeddings for most of the non-English languages. We use the model for inferencing using the HuggingFace Transformers library (Wolf et al., 2020).

3.2.1 Model Architecture

The model architecture of LaBSE is similar to BERT (Devlin et al., 2019) and uses self attention to process input text. This is then pre-trained on a large corpus that of multiple languages. After this pre-training the model can generate fixed length sentence embeddings. These embeddings are designed to be language-agnostic.

4 Experimental Setup

4.1 Dataset

We use the *SemRel* datasets (Ousidhoum et al., 2024a), a semantic relatedness dataset annotated by native accross 14 languages 14 languages: *Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish* and *Telugu*. These datasets consists of multiple records of sentence pairs along with their manually annotated relatedness score. All the languages of Track A of the dataset consist of a train-test split.

4.2 Embedding

We propose a system where we take a zero-shot approach to the test set with the pre-trained embedding models. We don't train or fine-tune the models used on the training data. We use the training data for evaluation of model performance on the languages in this track. We can find the evaluation of the models on the training set in table 2. Based on this performance we use text-embedding-3-large for Algerian Arabic, Moroccan Arabic, English, and Kinyarwanda and LaBSE for Amharic, Spanish, Hausa, Marathi and Telugu.

5 Results

For evaluation, the organizers rank the system based on Spearman rank correlation coefficient with the golden labels. The performance of the models on all the languages can be found in Table 3. Our system to identify the relatedness scores uses a zero shot method and achieves scores similar to the baseline scores. The score for English surpasses the baseline score.

6 Conclusions and Limitations

In our paper for the SemEval Task 1: Semantic Textual Relatedness we propose a zero-shot approach for relatedness using the text-embedding-3-large and LaBSE embedding models. It is importand to consider that text-embedding-3-large is not an open-source model and that these models may contain inherent biases in them.

A Spearman Correlation on test dataset

Language	Our scores
Algerian Arabic (arq)	0.5097117963
Amharic (amh)	0.8000962937
English (eng)	0.8323738277
Hausa (hau)	0.5083993463
Kinyarwanda (kin)	0.5183340316
Marathi (mar)	0.8415291711
Moroccan Arabic (ary)	0.4441887719
Spanish (esp)	0.6557116114
Telugu (tel)	0.814199637

Table 3: Spearman Correlation on test dataset

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new

perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Stephen A. Bernhardt. 1980. *Style*, 14(1):47–50.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Robert M Fano and WT Wintringham. 1961. Transmission of information.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding.

Zellig S. Harris. 1954. Distributional structure.

Martin Joos. 1950. Description of language design. *Journal of the Acoustical Society of America*, 22:701–707.

Hans Peter Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Saif Mohammad. 2008. *Measuring semantic distance using distributional profiles of concepts*. University of Toronto.

Saif M. Mohammad and Graeme Hirst. 2012. Distributional measures of semantic distance: A survey.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Charles E. Osgood, George J. Suci, and Percy H. Tannenbaum. 1957. *The Measurement of Meaning*. University of Illinois Press.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

SmurfCat at SemEval-2024 Task 6: Leveraging Synthetic Data for Hallucination Detection

Elisei Rykov[1,2], Yana Shishkina[2,3], Kseniia Petrushina[1,4], Kseniia Titova[1,5], Sergey Petrakov[1], and Alexander Panchenko[1,6]
[1]Skolkovo Institute of Science and Technology, [2]Tinkoff,
[3]HSE University, [4]Moscow Institute of Physics and Technology, [5]MTS AI, [6]AIRI
{e.rykov, y.a.shishkina}@tinkoff.ai, {kseniia.petrushina, kseniia.titova, sergey.petrakov, a.panchenko}@skol.tech

Abstract

In this paper, we present our novel systems developed for the SemEval-2024 hallucination detection task. Our investigation spans a range of strategies to compare model predictions with reference standards, encompassing diverse baselines, the refinement of pre-trained encoders through supervised learning, and an ensemble approaches utilizing several high-performing models. Through these explorations, we introduce three distinct methods that exhibit strong performance metrics. To amplify our training data, we generate additional training samples from unlabelled training subset. Furthermore, we provide a detailed comparative analysis of our approaches. Notably, our premier method achieved a commendable 9th place in the competition's model-agnostic track and 17th place in model-aware track, highlighting its effectiveness and potential.

1 Introduction

Large language models are proficient in generating human-like text across various styles. However, even the most advanced models can produce hallucinations, leading users to question their reliability. There are two primary types of hallucinations: factuality hallucinations, which involve the generation of content that deviates from actual facts, and faithfulness hallucinations, when the model fails to solve tasks correctly following specific instructions (Huang et al., 2023).

The SemEval 2024 Shared-task on Hallucinations and Related Observable Overgeneration Mistakes (Mickus et al., 2024) has integrated both types into three tasks. The Definition Modeling task (DM) focused on fact-related hallucinations by challenging models to generate contextually relevant word definitions. Both the Machine Translation (MT) and Paraphrase Generation (PG) tasks included faithfulness hallucinations, with models asked to produce translations or paraphrases for given sentences. Evaluation labelled datasets for these tasks were provided and the training dataset consisted only of source sentences and model generations, without corresponding labels.

Motivated by the lack of annotated resources and the efficacy of other language models trained on synthetic data, we developed two synthetic datasets that replicate the targeted domain. First, we collected data through a proprietary GPT-4 model (OpenAI, 2023), but our methods trained on the achieved data did not yield the desired results as prompt engineering made maintaining the domain challenging. As a second approach, we trained LLaMA2-7b (Touvron et al., 2023) adapters using a small set of annotated examples and applied them to the unlabeled training data. This method proved to be a more effective form of in-domain data augmentation.

While the competition was run on two tracks, we focus mainly on the model-agnostic track. In our methods we utilized the most effective models with varied sizes and architectures, which we had evaluated beforehand. Our experiments involved fine-tuning a pre-trained embedding model, repurposing it to function as a binary classifier across a number of open-source datasets, including our synthetic sets. We also experimented with a promising method for evaluating paraphrases by modifying its design and fine-tuning the model on different data. Finally, we tested different combinations of the highest-performing approaches in an ensemble setting. Generated synthetic data and code published on GitHub[1].

2 Related work

In the field of text representation, the E5 (Wang et al., 2022) family represents a group of cutting-edge sentence embedding models trained through

[1]https://github.com/s-nlp/shroom

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 869–880
June 20-21, 2024

contrastive methods. The E5-Mistral[2] model, a powerful embedding model that has been fine-tuned on a selection of annotated data, is currently recognized as the leading open-source model by the Multitask Text Embedding Benchmark (Muennighoff et al., 2023).

Vectara's *hallucination_detection_model*[3] is a fine-tuned DeBERTa focused on summarization datasets that includes annotations for factual consistency. TrueTeacher (Gekhman et al., 2023) is a family of models and an associated dataset designed for evaluating factual consistency. The dataset was created by first fine-tuning various-sized T5 models on summarization tasks. These models were then employed to generate hypotheses, which were subsequently automatically annotated using a 540B Large Language Model (LLM). This annotated dataset was then utilized to train multiple models to assess factual consistency.

The Mutual Implication Score (MIS) (Babakov et al., 2022) is a metric devised for evaluating the quality of text style transfer and paraphrasing systems, grounding its assessment on content similarity between the prediction and the reference text. It leverages a RoBERTa-NLI (Nie et al., 2020) model that has been fine-tuned and incorporates it into an architecture that processes two input texts sequentially in both forward and reverse directions. The final hidden states from these two passes are merged and forwarded to a classification head to determine the MIS score. Initially, the MIS metric was trained using the Quora Question Pairs dataset (QQP) (Sharma et al., 2019).

SimCSE (Similarity-based Contrastive Self-supervised Learning) (Gao et al., 2021) is a self-supervised learning method for text embeddings. It is used for creating embeddings of text data that are semantically meaningful and can be used in various downstream tasks. It involves training a neural network to maximize the similarity between embeddings of similar sentences and minimize the similarity between embeddings of dissimilar sentences. LaBSE (Language-agnostic BERT Sentence Embedding) (Feng et al., 2022) is a method for generating multilingual sentence embeddings using the BERT architecture.

Other metrics for evaluating content preservation, such as BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002), CHRF (Character n-gram F-score) (Popović, 2015), METEOR (Metric for Evaluation of Translation with Explicit ORdering) (Banerjee and Lavie, 2005), and BLEURT (Bilingual Evaluation Understudy for Ranking and Tuning) (Sellam et al., 2020), also stand out. BLEU utilizes a modified unigram precision score, CHRF evaluates the quality of machine translation by comparing character n-grams in candidate translations against reference translations to compute an F-score, METEOR calculates the harmonic mean of precision and recall at the single-word level, and BLEURT employs a fine-tuned BERT model in a cross-encoder setup, using synthetic data to assess semantic similarity.

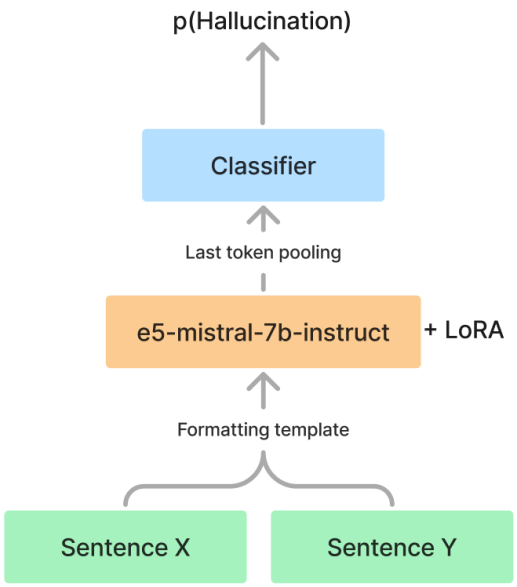


Figure 1: Classifier architecture when using synthetic data.

3 Data

3.1 Existing datasets

The QQP dataset consists of pairs of questions from the Quora forum. For each pair, it is indicated whether the questions are paraphrases, i.e. they ask about the same thing. PAWS (Zhang et al., 2019) is a paraphrase detection dataset that contains complex cases with both paraphrase and non-paraphrase samples that have high lexical overlap.

We postulated that other pre-existing datasets, such as QQP and PAWS, might exhibit particular biases due to their distinct task domains (for instance, QQP dataset includes only questions). To mitigate this potential issue, we generated synthetic

[2] https://huggingface.co/intfloat/e5-mistral-7b-instruct

[3] https://huggingface.co/vectara/hallucination_evaluation_model

	DM	MT	PG
Not Hallucination	188	211	132
Hallucination	175	179	132
Total	363	390	264

Table 1: Adapter train sample sizes.

data taking unlabeled training samples as starting points.

Our experimentation with synthetic data creation was divided into two main approaches: the first involved training LoRA (Hu et al., 2022) adapters for the LLaMA2-7b model using the annotated data derived from the validation set. The second approach involved the generation of both correct and incorrect hypotheses by employing GPT-4 and specific prompts.

All tasks were distilled down to the paraphrase evaluation task. Consequently, we only used targets (sources for paraphrase generation) and hypotheses as inputs for the models.

3.2 LLaMA2-7b adapter

We trained 6 LoRA adapters, pairing them to specialize in either generating hallucinations or producing correct responses for each task. Due to the limited amount of labeled data, we made use of model's ability of in-context learning by prepending samples with instructions: *Paraphrase* for non-hallucinations and *Provide an incorrect paraphrase* for hallucinations. The number of samples for each adapter is shown in Table 1.

Training and generation hyperparameters are displayed in Table 2. For each task and label we manually selected the best epoch by analyzing a small set of generated samples. These checkpoints were further employed to synthesize hypotheses for their task's training set. A small sample of the generated data using LLaMA2-7b adapter is provided in the Appendix C.

3.3 GPT-4 prompting

In addition, we created two distinct prompts for the PG task. In these prompts, we directed GPT-4 to generate a paraphrase of a source sentence extracted from an unlabeled training sample. The nature of the paraphrase, whether it should contain hallucinations and overgeneration errors or not, was determined by the specific prompt we used.

We enriched the prompt structure for few-shot learning purposes, incorporating several illustrative

Stage	Hyperparameter	Value
Training	lr	4e-4
	warmup_steps	1
	optimizer	AdamW
	scheduler	linear
	LoRA alpha	16
	LoRA dropout	0.05
	LoRA r	16
	batch size	32
Inference	num_beams	3
	do_sample	true
	repetition_penalty	1.2
	top_k	50
	max_new_tokens	512

Table 2: Training and inference hyperparameters for LoRA adapters.

examples drawn from both the validation and trial data splits. Alongside each *incorrect* example, we included an explanation to clarify why the provided hypothesis did not meet the criteria.

Moreover, we tasked GPT-4 to execute its reasoning step-by-step: to iterate through several examples with accompanying explanations, and, by leveraging those explanations, to discern and select the most suitable paraphrase.

We utilized the *gpt-4-1106-preview* model, adhering to the default generation parameters stipulated by the OpenAI API service.

3.4 Data filtration

In the process of evaluating the synthetic data we generated, we encountered multiple issues that necessitated an extra layer of filtering:

- A number of the samples produced by the LLaMA2-7B model were excessively lengthy, containing up to 1024 tokens.
- The labeling of samples by the LLaMA2-7B as *Hallucination* was frequently incorrect. Samples designated as hallucinations were often devoid of any such content, and conversely, non-hallucination samples sometimes contained hallucinations.
- A peculiar pattern was observed in the DM task generations from LLaMA2-7B, where more than 9,000 samples started with the word *any* or *anything* denoting a biased starting point which may impact the diversity and neutrality required for effective training.
- In the subsets of synthetic data generated by GPT-4 and labeled as *Not Hallucination* the resulting examples were deemed too straightforward, potentially leading to a training dataset

that cannot robustly challenge and thereby improve the model's discriminatory capabilities.

To tackle the identified issues with the synthetic data, we adopted a systematic filtering methodology. We began by eliminating any hypothesis that exceeded a length of 200 tokens, ensuring the data remained succinct. For the samples that started with *any* or *anything*, we decided to limit the number to 500 to minimize bias.

With the aim of refining the data quality, we then annotated all the synthetic samples using MIS. We set specific thresholds for these MIS scores to filter the data further. In the subset containing hallucinations, we removed samples that had a score lower than 0.1 or higher than 0.5. For non-hallucinated samples, we only retained those with a score between 0.7 and 0.9. These score ranges were established empirically to ensure a balance between discernibility and ambiguity in both the hallucinated and non-hallucinated examples.

The number of samples generated using both synthetic methods, before and after the filtering stage, is given in Table 3. After generating the synthetic data, we performed several experiments with different combinations of synthetic data.

4 Methods

4.1 Black-box baselines

First, we started with an assessment of various baseline models that are detailed in Section 2, including a new addition, GPT-4. These baseline models were utilized as-is, in a *black-box* fashion, without any further fine-tuning specifically for our tasks.

For all models other than GPT-4, we employed the inference code available on the official HuggingFace Hub pages. For GPT-4, we created specific prompts for each task. Within these prompts, we instructed GPT-4 to methodically process the information and ascertain the presence of hallucinations within the sample. We provided all pertinent data (source, hypothesis, and, when available, target) within the prompt. It is important to note that the collection and evaluation of predictions were conducted strictly within the model-aware track. The prompt is available in Appendix A.

4.2 SFT E5-Mistral

The obtained synthetic data was used to fine-tune the E5-Mistral model on our domain. In our experiments, we adjusted the data inputs by adding or omitting certain subsets of synthetic data to create the final blend used for training. The choice of the E5-Mistral model as the foundation for our work was based on its superior performance compared to other models.

The design of our classifier is depicted in Figure 1. In simple terms, we prepare two sample sentences with a specific format and input them into an model with LoRA. Afterwards, we obtain the embedding of the last token and pass it to the classification head.

4.3 Mutual Implication Score

In this setup we experimented with some improvements to the original Mutual Implication Score model architecture. Even though MIS was already trained on a large amount of paraphrase detection data, QQP dataset biased to the questions. Therefore, we thought that we can fine-tune it to decrease this bias.

In Table 4 we present default training hyperparameters used for experiments with MIS. Unless stated otherwise, we chose to train with the RoBERTa encoder, classifier and QQP dataset from original MIS study.

We tried various experiment configurations, ranging from the use of new datasets to alterations in architecture and training methods. We will describe all the modifications presented:

1. **MIS**: Vanilla MIS from HuggingFace Hub without any fine-tuning.
2. **MIS trained with LoRA**: Add LoRA adapters instead of partially unfreezing layers.
3. **MIS with Vectara**: Replace the original RoBERTa encoder with Vectara's model.
4. **MIS with one encoder**: Change MIS two-folded architecture with a single one.
5. **MIS trained on the PAWS**: Add 108,463 human-labeled paraphrase adversaries from PAWS.
6. **MIS trained on our synthetic data**: Add our synthetic data obtained previously.

4.4 Content Preservation Measures

We conducted a separate analysis on several NLP techniques as examined in the original MIS study. This exploration aimed to assess their suitability for the task of hallucination detection, considering the inherent connection between style transformation, paraphrase generation, and hallucination detection. A well-executed paraphrase should retain the essence of the original text without introducing

Source method	Task	Label	# before filtering	# after filtering
LLaMA2-7B	MT	Hallucination	18 093	7 758
		Not Hallucination	17 056	3 572
	PG	Hallucination	13 961	2 839
		Not Hallucination	14 928	3 952
	DM	Hallucination	19 224	5 939
		Not Hallucination	20 000	12 032
GPT-4	PG	Hallucination	7 439	-
		Not Hallucination	6 279	-

Table 3: The number of samples in the synthetic datasets. No filtering was performed for GPT-4.

Hyperparameter	Value
lr	1e-4
lr scheduler	constant
optimizer	AdamW
batch size	32

Table 4: Training hyperparameters for MIS experiments.

extraneous elements, which is particularly crucial given that one of the competition's subtasks involved paraphrasing. Specifically, our investigation involved LaBSE, SimCSE, and the metrics for evaluating content preservation described in Section 2.

4.5 Ensembling

To enhance the performance of different pre-trained models, we combined them into an ensemble. The final decision on the presence of hallucinations is based on the predictions of multiple independent models.

The predictions of separate models were normalized so that the decision boundary was the same for all models. Thus, differences in the scale of the threshold value did not introduce bias into the final decision.

We have chosen the best set of models for the ensemble from the possible options: E5-Mistral, fine-tuned E5-Mistral, Vectara, TrueTeacher, *all-mpnet-base-v2*[§] and also Mutual Implication Score. We calculated cosine between the encoded representations of the model's hypothesis and the target sentence. To obtain a prediction, this score was compared with a descision boundary. For each model we select the optimal classification threshold on validation subset for each track and task. For Vectara we used a threshold of 0.5.

[§]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

We employed different strategies on aggregating individual hallucination scores: Normalized averaging and Voting.

4.5.1 Normalized averaging

The predictions of separate models were normalized so that the decision boundary was the same for all models. Thus, differences in the scale of the threshold value did not introduce bias into the final decision.

Individual model scores are normalized as follows:

$$\hat{p} = \begin{cases} kp + b, & p \geq \text{thr} \\ \frac{p}{2\text{thr}}, & p < \text{thr} \end{cases}$$

where $k = \frac{1}{2(1-\text{thr})}$, $b = 1 - k$ and thr is the optimal decision boundary on validation.

This transformation allows to keep the score within $[0, 1]$, at the same time, the decision boundary for all models becomes 0.5.

4.5.2 Voting

Another strategy is to aggregate the binary predictions of the models in an ensemble. The presence of hallucinations was determined by voting models, depending on the number of votes in favor. At the verification stage, we determine the minimum number of model votes required to acknowledge the pair of sentences, model hypothesis and ground truth, as a paraphrase, for example, at least one, two or three models voted in favor. That is, we predicted a hallucination if an insufficient number of models compared to the optimal validation threshold classified the sample as a paraphrase.

5 Results

The comparative analysis of the performance across all baselines, our proposed methods, and the leading approaches derived from the official rankings is collated in Table 5.

Method	val		test	
	agnostic	aware	agnostic	aware
ahoblitz*	-	-	**0.85**	**0.81**
zackchen*	-	-	0.84	**0.81**
liuwei*	-	-	0.83	0.80
Voting	0.85	0.82	0.82	0.78
Normalized averaging	0.81	0.81	0.81	0.79
MIS + PAWS	0.82	0.82	0.81	0.78
SFT E5 Mistral	0.83	0.77	0.80	0.77
MIS	0.81	0.78	0.80	0.77
E5 Mistral	0.81	0.80	0.76	0.78
Vectara	0.76	0.76	0.75	0.77
TrueTeacher	0.79	0.79	0.76	0.80
GPT-4	-	0.74	-	-
SimCSE	0.80	0.80	0.76	0.76
BLEURT	0.77	0.77	0.74	0.74
LaBSE	0.72	0.75	0.69	0.73
METEOR	0.68	0.71	0.67	0.69
chrF	0.63	0.72	0.65	0.67
BLEU	0.67	0.70	0.64	0.65
Official baseline	-	-	0.70	0.74

Table 5: Performance of described approaches. Accuracy is observed as evaluation score. *Top approaches from the official rankings.

Method	Models	val		test	
		agnostic	aware	agnostic	aware
Voting	MIS + E5-Mistral + SFT E5-Mistral + all-mpnet + Vectara	**0.85**	**0.82**	**0.82**	0.78
	MIS + E5-Mistral + SFT E5-Mistral + all-mpnet	**0.85**	0.80	**0.82**	0.77
Normalized averaging	MIS + E5-Mistral + SFT E5-Mistral	**0.85**	0.79	0.81	0.78
	MIS + all-mpnet + Vectara + TrueTeacher	0.81	0.81	0.81	**0.79**

Table 6: Ensembling results. Accuracy is observed as evaluation score.

5.1 Ensembling

According to the results, the Voting approach we developed surpasses all baselines as well as other methods we devised. Nevertheless, the performance narrowly trails the foremost methods from the model-agnostic track in the official rankings by a minimal margin of 0.01. In regards to the application of Ensembling methods, a detailed evaluation delineating the constituent models employed is documented in Table 6. It was discerned that the incorporation of our SFT E5-Mistral model enhances overall performance metrics.

5.2 MIS

Succeeding in performance ranking is the MIS model, refined through training on the PAWS dataset. As previously elucidated, an assortment of configurations was examined, the details of which are exhaustively represented in Table 7. It is observed that the original MIS model's performance was not substantially uplifted; modifications yielded no marked increment in accuracy. Nonetheless, it is notable that the integration of the PAWS dataset into the training process marginally amplified accuracy for both tracks. Simultaneously, a minor enhancement on the aware track was observed upon the deployment of the Vectara encoder in place of the RoBERTa model.

5.3 SFT E5-Mistral

The next approach by performance is our SFT E5-Mistral. The accuracy for different configurations in our synthetic data experiments can be found in Table 8. The combination of PG and DM synthetic data achieves the best results. Unexpectedly, the use of synthetic data from GPT-4 does not yield as good outcomes. This suggests that GPT-4's synthetic data may contain some inherent biases.

Method	val		test	
	agnostic	aware	agnostic	aware
MIS (original)	0.80	0.78	0.77	**0.80**
+ LoRA	0.79	0.79	0.78	**0.80**
+ Vectara	0.79	0.81	**0.81**	0.77
+ Single fold	0.78	0.77	0.75	0.78
+ PAWS	**0.82**	**0.82**	**0.81**	0.78
+ Synthetic data	0.79	0.77	0.77	0.74

Table 7: MIS ablation study results. Accuracy is observed as evaluation score.

Source	Subset	agnostic	aware
GPT	PG	0.76	0.72
LLaMA	PG	0.81	0.75
	DM	0.63	0.51
	MT	0.79	0.71
	PG + DM	**0.83**	**0.77**
	PG + MT	0.81	0.76
	MT + DM	0.75	0.71
	All	0.77	0.71
GPT + LLaMA	All	0.77	0.73

Table 8: Synthetic data ablation study on E5-Mistral. Accuracy is observed as evaluation score.

We carried out a detailed evaluation of a particular subset and identified probable causes for bias:

- For texts generated without hallucinations, they tend to be overly formal and intricate.
- In cases with hallucinations, numerous instances are exceedingly convoluted, sometimes to the extent that the sentences convey the opposite meaning. Our investigation revealed that such hallucinations might not be readily detectable.

It is also clear that relying solely on DM synthetic data does not sufficiently address other tasks. By contrast, a model checkpoint trained with PG synthetic data shows promising performance. Just like the MIS approach, it appears that having PG data is sufficient to address hallucinations in other tasks, provided that the target is accessible.

5.4 Black-box baselines

All our advanced methods outperform black-box baselines on model-agnostic track. Even though, we observe that the E5-Mistral and MIS methods sets a solid baseline on model-agnostic track, maintaining a high level of performance even without any fine-tuning. Considering model-aware track, all baseline models except of GPT-4 show similar performance. The GPT-4 model does not do as well as the others in terms of the average score with our specific prompts. Finally, there is the official baseline that our approaches outperform.

5.5 Content Preservation Measures

Across preservation measures, SimCSE demonstrates the most notable results. In the model-agnostic track, it performs at the same level as more sophisticated approaches such as TrueTeacher, Vectara, or E5 Mistral, without any fine-tuning. However, other preservation measures do not perform as well. Most of them, with the exception of BLEURT, perform even worse than the official baseline in the model-agnostic track.

6 Conclusion

We conducted a comparative analysis involving six baseline models (MIS, E5-Mistral, Vectara, TrueTeacher, GPT-4, and the official baseline from the participant kit) alongside four sophisticated approaches (Voting and Normalized Averaging in Ensembling, as well as the refined MIS and SFT E5-Mistral). Of all methods evaluated, Ensembling demonstrated the highest performance. Nonetheless, the refined MIS and the SFT E5-Mistral exhibited only a minor shortfall in performance when compared to these leading methodologies.

Indeed, there appear to be several avenues for enhancing our synthetic data to potentially exceed the performance of other methods:

- Instead of training separate adapters for each task, centralized training with one adapter across multiple tasks could enrich the learning context and expand the size of the training dataset.
- Exploring a range of other models, such as Mistral-7b (Jiang et al., 2023), Mixtral-8x7b[¶], or LLaMA models of various larger sizes (LLaMA-13b, LLaMA-30b), could identify more efficient architectures or models that are better suited to handle the synthetic data effectively.
- For improving the quality of GPT-generated synthetic data, incorporating a more extensive range of examples within few-shot prompts and providing detailed explanations for the *correct* samples could help in mitigating bias and increasing the fidelity of the generated

[¶] https://huggingface.co/mistralai/Mixtral-8x7B-v0.1

data.

The potential use of our adapters to generate both positive and negative samples aimed at a specific target is indeed promising. By assembling datasets that offer these contrasting examples, we could refine the training process through contrastive fine-tuning. Such a method is hypothesized to yield superior performance by facilitating the model's ability to discern and learn from the nuanced differences between correct and incorrect instances.

References

Nikolay Babakov, David Dale, Varvara Logacheva, and Alexander Panchenko. 2022. A large-scale computational study of content preservation measures for text style transfer and paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 300–321, Dublin, Ireland. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023. TrueTeacher: Learning factual consistency evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2053–2070, Singapore. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2006–2029. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4885–4901. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset. *ArXiv*, abs/1907.01041.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *CoRR*, abs/2212.03533.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A GPT-4 prompt for PG task evaluation

```
Read the source sentence and the paraphrased hypothesis and answer whether there are any hallucinations
or related observable overgeneration errors for the paraphrasing task.
Before answering, think step by step and write why you chose the answer you did.
Answer the last string with 'The hypothesis is correct' if there are no hallucinations or misgenerations.
Otherwise, answer with 'The hypothesis is false'.

Example 1:
Source sentence: "The European Parliament does not approve the budget."
Paraphrased hypothesis: "The budget cannot be adopted against the will of the European Parliament."
The hypothesis is false

Example 2:
Source sentence: "Everyone is capable of enjoying a good education in a society."
Paraphrased hypothesis: "We must create a society where everyone is able to enjoy a good education."
The hypothesis is correct
```

Figure 2: Prompt for GPT-4 evaluation on PG task.

B GPT-4 prompt for synthetic paraphrased data generation with hallucinations

```
Your aim is to produce an incorrectly paraphrased sentence that contains a hallucination for the given source sentence. Hallucinations in a paraphrase can add new information that wasn't present in the source sentence, or exclude some important information, or reverse the meaning of the source sentence. Remember that reversing source sentence has the lowest level of priority, so use it only if there is no other way to make a hallucination. Usually it's much better to misrepresent some information, add new or exclude something important. If there is some quantitative information in the source, feel free to change them slightly. Complete the task using the examples below. The examples also show the correct paraphrase for the source sentences. Note that there are no hallucinations in the correct paraphrase, whereas your aim is to corrupt the source and produce a false paraphrase.

Examples:
Source: "I have a permit."
The correct paraphrase: "Uh, I'm validated."
The incorrect paraphrase: "I have a permit to carry it."
Explanation: The incorrect paraphrase adds information that is not present in the source sentence ("to carry it")

Source: "Easy, easy."
The correct paraphrase: "Watch it now."
The incorrect paraphrase: "The process is easy."
Explanation: The incorrect paraphrase introduces additional information ("The process is")

Source: "A five, six, seven, eight."
The correct paraphrase: "And 5, 6, 7, 8."
The incorrect paraphrase: "A number between five and eight."
Explanation: While the source sentence is a rhythmic count or sequence of specific numbers, the incorrect paraphrase generalizes it to "a number between five and eight".

Source: "A lot safer that way."
The correct paraphrase: "Because it's safer."
The incorrect paraphrase: "That is a safer way to travel."
Explanation: The major hallucination lies in the addition of "That is," which wasn't present in the original source sentence. This introduces a new element and changes the focus from the general concept of safety to a specific way of travel

Source: "You're a scam artist."
The correct paraphrase: "You are an imposter."
The incorrect paraphrase: "You're not a good scam artist."
Explanation: While the source sentence simply states "You're a scam artist," the incorrect paraphrase implies a judgment on the person's skill as a scam artist

Don't answer now, read the source and think step by step how to make a false paraphrase for the source sentence. Before answering, provide several examples with explanations and choose the best one. Answer starting with 'The incorrect paraphrase:
```

Figure 3: Prompt for PG data with hallucinations generation using GPT-4.

C Synthetic data examples

Task	Target	Hypothesis	Label
	I know you kissed Tom.	I know you're going to beat someone.	Hallucination
	I'll talk to Tom today.	I'll talk to John tomorrow.	Hallucination
	I should've waited.	I'll be better tomorrow.	Hallucination
MT	He was mistaken for his younger brother	He was taken for his brother.	Not Hallucination
	She passed the examination.	She successfully completed the certification.	Not Hallucination
	A plane popped up out of nowhere.	A plane just came out of the blue.	Not Hallucination
	No one has seen Tom since.	No one has seen Tom for a long time.	Not Hallucination
	Wanna talk?	Would you want to talk with me?	Not Hallucination
	We must not allow this to happen.	We must prevent this from happening.	Not Hallucination
	Have you found her?	Have you seen her?	Hallucination
PG	A word of advice.	Give me some advice.	Not Hallucination
	Hold your course.	You're going the wrong way, man.	Hallucination
	Can I take a message?	Can I take a message for you,	Not Hallucination
	My job?	My job is to carry out the trash.	Hallucination
	Delicious .	(scrambley) A scrambley dish.	Hallucination
	To increase the level or amount of .	To increase in volume.	Not Hallucination
	Causing the air to be hot .	Hot. Something that is hot.	Not Hallucination
DM	(slang, derogatory) schizoid, schizophrenic; crazy	(transitive) Crazy	Not Hallucination
	Covered with petals or petal-like objects.	planted.	Hallucination
	Alternative form of midstream	Middle stream	Not Hallucination
	To require	take time to finish something.	Hallucination

Table 9: Sample of synthetic data generated using LLaMA2-7B

Target	Hypothesis	Label
That cannot be in our interest!	It's not beneficial for us!	Not hallucination
The written language should be made more user-friendly.	The spoken language should be made more user-friendly.	Hallucination
I do not think that is quite what the agreement is.	I do not think that's the contract we signed.	Hallucination
The vote will take place tomorrow at 11.30 a.m.	Tomorrow, the voting process is scheduled for 11.30 in the morning.	Not hallucination
Mrs Green, you have the floor.	Mrs. Green, you own the flooring.	Hallucination
I was also in a northern industrial suburb in Milan.	I too have been to one of Milan's northern industrial neighborhoods.	Not hallucination
Mr President, I should like to make a further remark.	Mr. President, I would like to add another comment.	Not hallucination
Mrs Bonino tells me that no response is necessary.	Mrs. Bonino informed me a response isn't required.	Not hallucination

Table 10: Sample of synthetic data generated using GPT-4

USTCCTSU at SemEval-2024 Task 1: Reducing Anisotropy for Cross-lingual Semantic Textual Relatedness

Jianjian Li[1 *] **Shengwei Liang**[1 *] **Yong Liao**[1,2 †]
Hongping Deng[2] **Haiyang Yu**[2†]
1. University of Science and Technology of China, CCCD Key Lab of MCT
2. Institute of Dataspace
{sa22221088,sewell}@mail.ustc.edu.cn

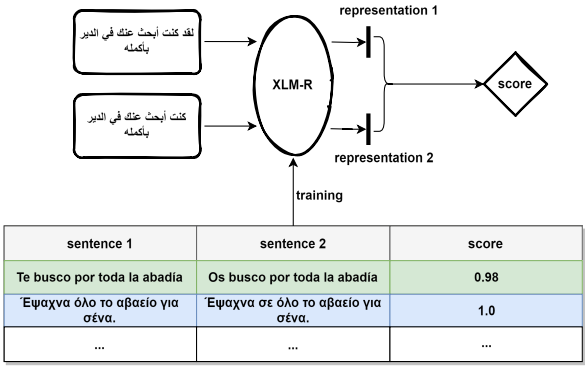
Abstract

Cross-lingual semantic textual relatedness task is an important research task that addresses challenges in cross-lingual communication and text understanding. It helps establish semantic connections between different languages, crucial for downstream tasks like machine translation, multilingual information retrieval, and cross-lingual text understanding. Based on extensive comparative experiments, we choose the $XLM\text{-}R_{base}$ as our base model and use pre-trained sentence representations based on whitening to reduce anisotropy. Additionally, for the given training data, we design a delicate data filtering method to alleviate the curse of multilingualism. With our approach, we achieve a **2nd** score in Spanish, a **3rd** in Indonesian, and multiple entries in the top ten results in the competition's track C. We further do a comprehensive analysis to inspire future research aimed at improving performance on cross-lingual tasks.

1 Introduction

Semantic textual relatedness (STR) encompasses a broader concept that takes into account various commonalities between two sentences. This includes factors such as being on the same topic, expressing the same viewpoint, originating from the same period, one sentence elaborating on or following from the other, and more. SemEval is an international workshop on semantic evaluation. In track C of SemEval-2024 task 1: Cross-lingual (Ousidhoum et al., 2024b), participants are to submit systems,which are developed without the use of any labeled semantic similarity or semantic relatedness datasets in the target language and with the use of labeled datasets (Ousidhoum et al., 2024a) from at least one other language.

Various methods were proposed to address the task of textual relatedness. One common approach



sentence 1	sentence 2	score
Te busco por toda la abadía	Os busco por toda la abadía	0.98
Έψαχνα όλο το αβαείο για σένα.	Έψαχνα σε όλο το αβαείο για σένα.	1.0
...	...	...

Figure 1: The description of cross-lingual semantic textual relatedness task.

is based on feature engineering, where the syntactic, semantic, and structural features of text, such as word frequency, TF-IDF, and word embeddings, are extracted. Machine learning algorithms are then employed for relatedness determination. Another popular approach is based on deep learning methods, such as Convolutional Neural Networks (LeCun et al., 1989), Recurrent Neural Networks (Graves and Graves, 2012) , and self-attention mechanisms (Vaswani et al., 2017). These methods can capture semantic relationships and contextual information within the text, and they are trained on large-scale datasets to enhance model performance and generalization ability.

However, there are two challenges in track C of SemEval-2024 task 1:

- Compared with static word representation such as Word2Vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014), the pre-trained language models (PLM) can obtain sentence representation for different sentence in different contexts, thereby solving different problems. However, the vectors of BERT-based PLM models have limitations: **BERT-based models always induces a non-smooth anisotropic semantic space of sentences,**

[*] Equal contribution and shared co-first authorship.
[†] Corresponding author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 881–887

which harms its performance of semantic similarity (Gao et al., 2019; Li et al., 2020), which can lead to a challenge that sentences are strikingly similar while using the cosine similarity metric.

- Participants are not allowed to utilize labeled datasets in the target language for training. Instead, they must use labeled data in different languages as the training set to train the model and provide predictions in the target language. However, **multilingual pre-trained models suffer from the curse of multilingualism** (Conneau et al., 2020), that is, the overall performance of both monolingual and cross-lingual baselines declines when adding more languages to training data over a certain point. Hence, it is essential to investigate which additional languages would be inefficient as the training dataset for the target language.

In this paper, we used whitening techniques (Su et al., 2021), which maps vectors to standard orthogonal bases, to transform the word vector representations from anisotropic to isotropic, and surprisingly, we found that whitening significantly improves the accuracy of judging semantic similarity. Given the absence of labeled data in the target language, it is difficult to determine which other language would yield better prediction results when used as training data. Therefore, we proposed that removing certain language categories from the training data for a specific target language contributed to improving performance.

We conducted extensive experiments to demonstrate the effectiveness of the method we employed. As a result, our submitted outcomes achieved a **2nd** score in Spanish and a **3rd** score in Indonesian in track C of SemEval-2024's task 1. Additionally, we obtained multiple top-ten rankings in the competition.

2 Background

The task of semantic text relatedness covers several specific subtasks, including semantic similarity, semantic matching, textual entailment, semantic relation classification, and text pair ranking. Previous work has proposed various methods for these specific tasks, such as: Lexical and syntactic-based methods (Gamallo et al., 2001; Pakray et al., 2011): These methods rely on lexical and syntactic rules, such as word vector matching, lexical overlap, and syntactic tree matching. However, these methods often fail to capture higher-level semantic relationships. Feature engineering-based machine learning methods (Chia et al., 2021; Fan et al., 2019): These methods involve using manually designed features, such as bag-of-words models (Zhang et al., 2010), tf-idf weights, and syntactic features, followed by using machine learning algorithms like support vector machines and random forests for prediction.

While these methods have improved performance to some extent, they still have limitations in capturing complex semantic relationships. Neural network-based models: These models use neural networks to learn representations of text and capture semantic relationships between texts through training data. This includes methods that fine-tune pre-trained language models (e.g., BERT (Kenton and Toutanova, 2019) and GPT2 (Radford et al., 2019) etc.), as well as approaches that employ Siamese networks, LSTM, CNN, and other architectures for text encoding and matching. Transfer learning and multi-task learning (Pilault et al., 2020; Wu et al., 2020): These methods leverage knowledge from pre-trained models on related tasks to improve the performance of semantic textual relatedness tasks through transfer learning (Koroleva et al., 2019). Multi-task learning combines multiple related tasks in training to enhance the model's generalization ability and effectiveness. Application of external knowledge resources: Researchers have also attempted to incorporate external knowledge resources such as word embeddings, semantic knowledge graphs, and multilingual data to enhance the model's understanding of semantic relationships.

For cross-lingual semantic similarity tasks, mapping texts from different languages into a shared semantic space for similarity calculation is necessary. To address this, researchers have proposed various cross-lingual representation learning methods. Among them, unsupervised alignment methods like unsupervised machine translation (Lample et al., 2017) and cross-lingual pre-training models (Liang et al., 2020) can learn the correspondences between multiple languages and map texts to a shared vector space.

However, (Conneau and Lample, 2019) and (Wang et al.) mentioned that vector representations based on the Transformer models exhibit anisotropy, which means that the vectors are unevenly distributed and clustered in a narrow cone-shaped space. Therefore, both Bert-flow (Li et al.,

2020) and Bert-whitening (Su et al., 2021) aim to address the same issue, which is the anisotropy and uneven distribution of sentence embeddings.

3 System Overview

3.1 Framework Overview

In this section, we will introduce our proposed method for STR task which has three main modules.

- **PLM Encoder** We adopted the pretrained language model XLM-RoBERTa-base ($XLM\text{-}R_{base}$) (Conneau et al., 2020) for initial sentence encoding, which combines two powerful models: Transformer and RoBERTa. $XLM\text{-}R_{base}$ demonstrates strong multilingual capabilities and a deep understanding of semantics, surpassing some monolingual pre-training models. After conducting a series of tests on mBERT (Pires et al., 2019), XLM (Conneau and Lample, 2019), and $XLM\text{-}R_{base/large}$, we selected $XLM\text{-}R_{base}$ as the encoder due to its superior performance.

- **Whitening Module** After obtaining the sentence vectors of two utterances using $XLM\text{-}R_{base}$, we could have directly calculated the cosine similarity between the two vectors, but the sentence vectors after $XLM\text{-}R_{base}$ show anisotropy between them and are distributed in a conical space, resulting in a high cosine similarity. Therefore, we introduce the Whitening module to change the distribution of the sentence vector space so that its distribution has various anisotropies, amplifying the differences between the vectors and stimulating the performance of $XLM\text{-}R_{base}$ on the semantic text similarity reading task.

- **Data Filtering** The authors of (Conneau et al., 2020) mention the curse of multilingualism, where adding more languages leads to an improvement in cross-lingual performance for low-resource languages up to a certain point, after which the overall performance of both monolingual and cross-lingual baselines declines. In the task of cross-lingual semantic text similarity, to maximize the exploration of the positive impact of other languages on the target language, we propose a new dataset selection method. As the influence between languages is mutual, we utilize the unlabeled data of the target language to detect the impact of each language in track A, excluding the target language, and infer its influence on the target language. This allows us to select the training dataset optimally. This approach helps eliminate interference from certain languages on the target language and avoids the curse of multilingualism.

3.2 PLM Encoder

Through a simple test and comparative analysis of different multilingual pre-training models, we found that $XLM\text{-}R_{base}$ outperforms mBERT. $XLM\text{-}R_{base}$ is a cross-lingual pre-training model based on the BERT architecture, an improvement and extension of the original XLM model. The goal of $XLM\text{-}R_{base}$ is to enhance the performance and effectiveness of multilingual text processing. $XLM\text{-}R_{base}$ utilizes larger-scale pre-training data and more sophisticated training methods to enhance the model's representation capabilities. It undergoes deep learning on a large amount of unsupervised data using RoBERT (Liu et al., 2019) technology. This enables $XLM\text{-}R_{base}$ to better understand and capture the semantic and grammatical features between different languages. Compared to the original XLM, $XLM\text{-}R_{base}$ has made several improvements. Firstly, it introduces a dynamic masking mechanism that allows the model to better perceive contextual information. Secondly, $XLM\text{-}R_{base}$ emphasizes cross-lingual consistency learning through adversarial training, enabling better alignment and sharing of model parameters. This enables $XLM\text{-}R_{base}$ to provide more accurate representations of texts in cross-lingual tasks. Compared to mBERT, $XLM\text{-}R_{base}$ employs larger-scale pre-training data, covers more languages, and incorporates improvements through RoBERTa technology. This enables $XLM\text{-}R_{base}$ to better learn and capture the semantic and grammatical features between different languages, thereby enhancing the model's representation capabilities and performance.

3.3 Whitening Module

Due to the existence of anisotropy among the vectors obtained from the initial encoding by $XLM\text{-}R_{base}$, cosine similarity cannot accurately measure the semantic similarity between sentences. Therefore, we chose to use whitening to map the original vector space to an isotropic space, where the vectors are transformed into vectors in a standard

orthogonal bases. The principle is as follows:

Suppose we have a set of sentence vectors $S = \{s_1, s_2, \ldots, s_n\}$, the set of vectors can be transformed into a set of vectors with isotropy (i.e., zero mean and a covariance matrix of the identity matrix) through the following transformation $\tilde{S} = \{\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n\}$.

$$\tilde{s}_i = (x_i - \mu)\mathbf{W} \tag{1}$$

If we want to make the set $\tilde{S}$ have a zero mean, we need to:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{2}$$

The next step is to calculate $\mathbf{W}$. The covariance matrix of S:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{s_i} - \mu)^\top (s_i - \mu) \tag{3}$$

The covariance matrix of $\tilde{S}$:

$$\tilde{\Sigma} = \mathbf{W}^\top \Sigma \mathbf{W} \tag{4}$$

If we want to transform $\tilde{\Sigma}$ into the identity matrix I, we need to:

$$\tilde{\Sigma} = \mathbf{W}^\top \Sigma \mathbf{W} = \mathbf{I} \tag{5}$$

Then:

$$\Sigma = (\mathbf{W}^\top)^{-1}\mathbf{W}^{-1} = (\mathbf{W^{-1}})^\top \mathbf{W}^{-1} \tag{6}$$

Since Σ is a positive definite symmetric matrix as the covariance matrix, it can be decomposed using Singular Value Decomposition (SVD), yielding:

$$\Sigma = \mathbf{U}\Lambda\mathbf{U}^\top \tag{7}$$

By combining equations (6) and (7), we obtain:

$$(\mathbf{W^{-1}})^\top \mathbf{W}^{-1} = \mathbf{U}\Lambda\mathbf{U}^\top = \mathbf{U}\sqrt{\Lambda}\sqrt{\Lambda}\mathbf{U}^\top \tag{8}$$

Then:

$$(\mathbf{W^{-1}})^\top \mathbf{W}^{-1} = (\sqrt{\Lambda}\mathbf{U}^\top)^\top \sqrt{\Lambda}\mathbf{U}^\top \tag{9}$$

Therefore, we can obtain $\mathbf{W^{-1}} = \sqrt{\Lambda}\mathbf{U}$, and finally obtain $\mathbf{W}$ as follows:

$$\mathbf{W} = \mathbf{U}\sqrt{\Lambda^{-1}} \tag{10}$$

3.4 Data Filtering

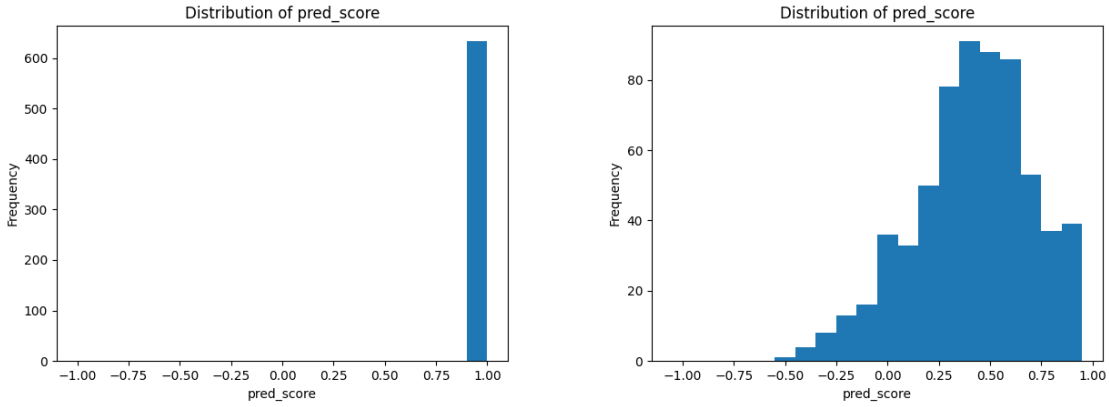
Our experiments have shown that when selecting training data for the target language, using a mixture of multiple languages often yields better results than using a single language. The authors of the $XLM\text{-}R_{base}$ paper mentioned that incorporating more languages improves the cross-lingual performance of low-resource languages up to a certain point. Beyond that point, the overall performance of both monolingual and cross-lingual benchmarks starts to decline. Additionally, we believe that there is interdependence between languages. For example, if including text from language A in training set to compute whitening parameters leads to a decrease in the prediction performance for language B, we expect that the opposite would hold true as well.

Therefore, inspired by this insight, we used the text in the target language as the dataset and individually tested the labeled training data provided in track A for different languages. For example, if the target language is identified by T, we use the text of T for whitening, and test the performance on language $Test_A$, $Test_B$, $Test_C$, $Test_D$,... one by one. If the prediction performance of $Test_A$ decreases after using T compared to not using T (measured by the Spearman correlation (Myers and Sirois, 2004) between the gold labels and predicted labels obtained using language $Test_A$), then $Test_A$ is excluded from target language's training set.

In the case of the Spanish, using the training set without data filtering (1,000 each of all data except Spanish) resulted in a final spearman coefficient of 0.6375; using the training set with data filtering (1000 each of kin and ind) resulted in a final spearman coefficient of 0.6886. Although the training data for about ten languages were reduced, the results were are significantly improved.

4 Experimental Setup

We use the 12 labeled training data from (Ousidhoum et al., 2024a) as training data and the test data from track C as test data. We observe that the amount of data for each language is concentrated around 1,000, so we take 1,000 as the boundary, use oversampling to make up for less than 1,000, and use randomization to take out 1,000 for more than 1,000 to ensure that sentence pairs of different similarities are involved. In finding the training set combinations for the target languages, we compute



(a) without whitening (b) with whitening

Figure 2: The results of model without whitening and with whitening.

the μ and $\mathbf{W}$ of whitening using the text data of the target languages in track C. We predict the training data one by one for each language, and compute the spearman coefficients using the gold labels and the predicted labels of the training data, and compare the results with the data without any whitening (i.e., the prediction result of the base model) to evaluate whether the target language enhances a certain language in the training data or not, and if it does not, it is excluded from the train data. Eventually, the remaining language data is used as a training set to predict the target language.

The hyperparameters are set as follows: we choose to freeze the pretrained model $XLM\text{-}R_{base}$ while setting the topk parameter of whitening to 256. The rubric we used was the spearman coefficient, calculated using the methodology provided by the competition officials.

5 Results

The official competition used the spearman coefficients to evaluate the results, and Table 1 gives the results of the spearman coefficients for both Indonesian (ind) and Spanish (esp) languages throughout the experiment. There is a big difference in the multilingual ability of different model bases. We chose $XLM\text{-}R_{base}$, which performs better, and we can see that the overall results are improved after using the whitening module to transform the vector space; $XLM\text{-}R_{base}$ with whitening is better than baseline, and we got a good ranking in track C of SemEval-2024 task 1, in which we ranked second in esp and third in ind.

As can be seen from Table 1, the whitening module improves the STR task more significantly,the

	ind-test	**esp-test**
Baseline	0.4700	0.6200
mBERT	0.4390	0.5971
$XLM\text{-}R_{base}$	0.4390	0.5907
$XLM\text{-}R_{large}$	0.4267	0.6003
mBERT-whitening	0.4471	0.6411
$XLM\text{-}R_{base}$-whitening	0.4746	**0.6886**
$XLM\text{-}R_{large}$-whitening	**0.4845**	0.6648

Table 1: The spearman coefficient of different models and baseline.

baseline is given by (Ousidhoum et al., 2024a). In order to further verify whether whitening works, we counted the cosine similarity distribution statistics of the data without whitening processing and after whitening. Figure 2 gives two cosine similarity statistics. The left side is the cosine similarity statistics without whitening. The cosine similarity of all utterance pairs is concentrated between 0.9 and 1.0, indicating that the vector space is anisotropic. In contrast, after adding whitening, the whole distribution tends to be normal, which indicates that whitening plays a role in mapping the vectors to an isotropic space, amplifying the differences between statements.

6 Conclusion

We use $XLM\text{-}R_{base}$ with whitening and propose a dataset filtering method that exploits the positive correlation of linguistic interactions, achieving good rankings in SemEval-2024 task 1 track C. We verifies that whitening performs well on utterance characterization as well as STR task. Besides, the proposed dataset filtering method is more efficient

and can alleviate the multilingual curse problem in cross-language problems to some extent.

In the future, we will further study this positive correlation of language interactions, and we hope that this correlation can become more detailed, not only in terms of inter-language correlations but also in terms of the domain of the text. We also hope that this correlation can be better utilized in dataset preprocessing, not only to eliminate poorly performing languages but to further improve the combination of datasets that can be directly selected to correspond to the optimal solution.

Acknowledgments

We want to express gratitude to the anonymous reviewers for their hard work and kind comments, which will further improve our work in the future. This work is funded by national key research and development program under grant 2021YFC3300500-02.

References

Zheng Lin Chia, Michal Ptaszynski, Fumito Masui, Gniewosz Leliwa, and Michal Wroczynski. 2021. Machine learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection. *Information Processing & Management*, 58(4):102600.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Cheng Fan, Yongjun Sun, Yang Zhao, Mengjie Song, and Jiayuan Wang. 2019. Deep learning-based feature engineering methods for improved building energy prediction. *Applied energy*, 240:35–45.

Pablo Gamallo, Caroline Gasperin, Alexandre Agustini, and Gabriel P Lopes. 2001. Syntactic-based methods for measuring word similarity. In *International Conference on Text, Speech and Dialogue*, pages 116–125. Springer.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. *arXiv e-prints*, pages arXiv–1907.

Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Anna Koroleva, Sanjay Kamath, and Patrick Paroubek. 2019. Measuring semantic similarity of clinical trial outcomes using deep pre-trained language representations. *Journal of Biomedical Informatics*, 100:100058.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. Xglue: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Leann Myers and Maria J Sirois. 2004. Spearman correlation coefficients, differences between. *Encyclopedia of statistical sciences*, 12.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata,

Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *Preprint*, arXiv:2402.08638.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Partha Pakray, Sivaji Bandyopadhyay, and Alexander Gelbukh. 2011. Textual entailment using lexical and syntactic similarity. *International Journal of Artificial Intelligence and Applications*, 2(1):43–58.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jonathan Pilault, Amine Elhattami, and Christopher Pal. 2020. Conditionally adaptive multi-task learning: Improving transfer learning in nlp using fewer parameters & less data. *arXiv preprint arXiv:2009.09139*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. Improving neural language generation with spectrum control.

Sen Wu, Hongyang R Zhang, and Christopher Ré. 2020. Understanding and improving information transfer in multi-task learning. *arXiv preprint arXiv:2005.00944*.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1:43–52.

GreyBox at SemEval-2024 Task 4: Progressive Fine-tuning (for Multilingual Detection of Propaganda Techniques)

Nathan Roll
University of California, Santa Barbara
nroll@ucsb.edu

Calbert Graham
University of Cambridge
crg29@cam.ac.uk

Abstract

We introduce a novel fine-tuning approach that effectively primes transformer-based language models to detect rhetorical and psychological techniques within internet memes. Our end-to-end system retains multilingual and task-general capacities from pretraining stages while adapting to domain intricacies using an increasingly targeted set of examples– achieving competitive rankings across English, Bulgarian, and North Macedonian. We find that our monolingual post-training regimen is sufficient to improve task performance in 17 language varieties beyond equivalent zero-shot capabilities despite English-only data. To promote further research, we release our code publicly on GitHub: github.com/Nathan-Roll1/GreyBox.

1 Introduction & Background

The digital age has radically transformed the nature of propaganda and disinformation, requiring innovative detection mechanisms attuned to these shifts (DeCook, 2018; Macdonald, 2006; Sparkes-Vian, 2019).

Previous work on propaganda detection (Li et al., 2019) leveraged a logistic regression model to determine whether or not a given passage was propagandistic using vectors based on Linguistic Inquiry and Word Count (LIWC), TF-IDF, BERT, and sentence features. These researchers have reported an F1 score of 66.16%, which significantly outperformed their baseline model. Oliinyk et al. (2020) used a similar architecture on the task, achieving improved performance by replacing manual feature selection with induced sentence-level and article-level vectors. Elhadad et al. (2020) used a variety of machine learning models, including logistic regression, to create an ensemble classifier for COVID-19 misinformation. More recently, there has been an emergence of work focusing on detection of propaganda in memes, with Dimitrov et al. (2021) releasing a corpus of memes, hand-labeled with one of 22 propaganda techniques, and utilizing a fusion of large language models (LLMs) to successfully identify labels for a shared task: "Multilingual Detection of Persuasion Techniques in Memes" (*SemEval 2024 Task 4*).

The purpose of the shared task is to foster the development of systems which detect rhetorical and psychological devices, often propagandistic in nature, from memes (a more comprehensive explanation is available in Dimitrov et al., 2024). It contains the following subtasks:

- *Subtask 1*: Given exclusively the text extracted from a given meme, identify the specific persuasion technique(s) utilized (if any).
- *Subtask 2*: Given both the text and image of a meme, identify the specific technique(s) being utilized (*Subtask 2a*), and whether or not the meme contains any propagandistic techniques (*Subtask 2b*).

Our system primarily tackles *Subtask 1*, using the text of a given meme to identify which, if any, of the devices are present. Our approach builds on Dimitrov et al. (2021), in tackling the challenge by leveraging the comprehensive pretraining of large language models (LLMs) and fine-tuning it with human-annotated examples.

The multilingual and multi-task capabilities of LLMs have been well established, however low-resource languages and tasks often require additional data to meet or exceed human level performance. Given that fine-tuning generally degrades baseline model capabilities (Zhai et al., 2023), this reality presents obstacles when available language data does not extend to desired task contexts or vice-versa. Through interative refinement, we discover that successive fine-tuning rounds – encompassing increasing task-specific data – result in models which better adapt to our specific task while

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 888–893

also retaining sufficient multilingual capabilities. Our approach to split the post-training regimen into multiple steps finds support in prior research. Xu et al. (2021) found that multi-stage fine-tuning has downstream benefits, particularly in low-resource settings. ValizadehAslani et al. (2022) examined the challenge of class imbalance by introducing a two-stage fine-tuning strategy in which they initially adjusted the model with a class-balanced 'reweighting' loss to ensure that underrepresented classes are not overlooked.

Our system makes use of the provided English meme data, manually labeled according to the requirements of the corresponding task. A total of 18,650 training examples generated from 11,111 unique memes were provided across the training, development, and validation splits.

This paper describes our system and explores how progressive fine-tuning learns the syntactic and semantic properties of memes, with potential future applications in a variety of tasks. For more details, please see the task paper Dimitrov et al. (2024).

2 System Overview

Our system leverages a novel, multi-stage fine-tuning process which progressively adapts a pre-trained LLM (GPT 3.5-Turbo[1]) to the task of identifying persuasion techniques in memes. This process consists of two distinct fine-tuning steps (see figure 1):

1. **Priming for meaning**: Expose the LLM to all released data in the train and validation splits to understand the context, intention, and implied meanings in memes.

2. **Structural adaptation**: Undergo an additional fine-tuning round on only *Subtask 1* data to align to the specific structural requirements of the output.

2.1 Data preparation

Each of the provided `.json` files were parsed into Python dictionaries, and reformatted into chat-like training examples with the text of the meme as the "user" and the label(s) as the "assistant"[2]. Memes which appeared in multiple subtask train/validation sets (based on the `id` field) were filtered to only include a single instance of each. The reformatted chat examples were saved as `.jsonl` files and programmatically uploaded to the OpenAI fine-tuning API[3] for usage.

Language	Rank	F_h	Pr_h	Rec_h
English	5/32	0.670	0.652	0.688
Bulgarian	7/19	0.476	0.438	0.521
N. Macedonian	8/19	0.434	0.440	0.430

Table 1: Official performance on *Subtask 1* languages.

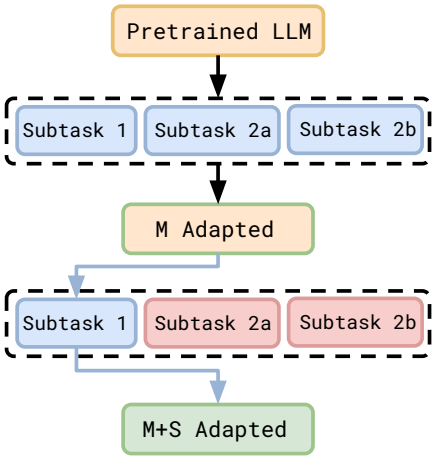


Figure 1: Our implementation of progressive fine-tuning on the *SemEval 2024 Task 4* data. Meaning-based (M) fine-tuning on broader data precedes a more targeted structural (S) fine-tuning step.

2.2 Fine-tuning

2.2.1 Step 1: Priming for Meaning

The priming stage of our fine-tuning process leveraged the train and validation splits across all subtasks. Given that each subtask has a distinct labeling methodology, the purpose of the priming stage is to impart task-specific knowledge (in terms of relevant tokens and their relationship to human-generated labels). Three epochs of fine-tuning were performed on GPT 3.5 Turbo with the training set, using the validation split to ensure that no overfitting was occurring during training. A total of 2.9M tokens were processed during the priming stage.

[1]For zero-shot evaluation, fine-tuning, and experiments we use the `gpt-3.5-turbo-1106` model with a context window of 16,385 tokens and a maximum output length of 4,096 tokens.

[2]We leave the system prompt blank in our fine-tuning pipeline to avoid excess costs from input redundancy.

[3]The GPT-3.5 family model weights can only be interacted with using OpenAI's API.

	Avg. F_h	GPT 3.5 Turbo			llama-2-70b-chat			mixtral-8x7b-instruct			Baseline[4]
		F_h	Pr_h	Rec_h	F_h	Pr_h	Rec_h	F_h	Pr_h	Rec_h	F_h
English	**0.276**	**0.281**	0.194	**0.512**	**0.270**	0.180	**0.538**	**0.277**	0.185	**0.556**	0.358
Spanish	0.265	0.275	**0.194**	0.470	0.257	0.176	0.481	0.264	0.179	0.503	" "
French	0.264	0.268	0.187	0.472	0.250	0.170	0.466	0.274	**0.186**	0.524	" "
Haitian Creole	0.258	0.259	0.193	0.393	0.256	0.181	0.438	0.259	0.176	0.492	" "
Ukrainian	0.257	0.265	0.189	0.443	0.246	0.166	0.469	0.261	0.176	0.500	" "
Turkish	0.253	0.264	0.190	0.432	0.239	0.165	0.432	0.257	0.176	0.478	" "
Finnish	0.253	0.264	0.192	0.426	0.231	0.160	0.414	0.263	0.180	0.488	" "
Chinese (Simp.)	0.251	0.259	0.184	0.439	0.243	0.168	0.441	0.251	0.172	0.461	" "
Chinese (Trad.)	0.251	0.265	0.191	0.436	0.246	0.172	0.435	0.241	0.166	0.440	" "
Swahili	0.250	0.250	0.183	0.395	0.237	0.166	0.418	0.262	0.181	0.476	" "
Hindi	0.248	0.254	0.183	0.415	0.250	**0.183**	0.397	0.239	0.160	0.469	" "
Arabic	0.246	0.264	0.188	0.445	0.233	0.174	0.352	0.241	0.165	0.447	" "
Yoruba	0.223	0.216	0.183	0.263	0.221	0.154	0.388	0.234	0.162	0.420	" "
Tamil	0.221	0.214	0.183	0.259	0.222	0.162	0.352	0.226	0.156	0.411	" "
Burmese	0.216	0.187	0.194	0.181	0.247	0.175	0.424	0.214	0.148	0.390	" "
Amharic	0.196	0.143	0.141	0.146	0.227	0.157	0.406	0.219	0.147	0.423	" "
Mean	*0.246*	*0.246*	*0.185*	*0.383*	*0.242*	*0.169*	*0.428*	*0.249*	*0.170*	*0.467*	***0.358***

Table 2: Zero-shot performance on the *Subtask 1* development set varies by model and source language.

2.2.2 Step 2: Structural Adaptation

Model finalization involved an additional two epochs of fine-tuning on the pragmatically-primed model, using only data specific to *Subtask 1*. Two epochs of training were performed, however we encourage further study on the impact of hyperparameters on downstream performance.

2.3 Evaluation Metrics

To capture the hierarchical nature of propaganda techniques, we utilize three metrics which weight errors based on their similarity to each other via higher order categories: hierarchical precision (Pr_h), hierarchical recall (Rec_h), and hierarchical F1 score (Pr_h) (Silla and Freitas, 2011). While the official evaluation of the task does not require leaf-node predictions, our system is not designed to output broader categories in cases of ambiguity. Further justification for the usage of these metrics, along with the exact hierarchy, is provided in Dimitrov et al. (2024).

2.3.1 Hierarchical Precision

Hierarchical precision (Pr_h) measures, in aggregate, the quality of each prediction. This metric is defined as the weighted sum of the predicted classes and their ancestors in the hierarchy, normalized by the total weight of the predicted classes across all test examples. It is given by:

$$Pr_h = \frac{\sum_i |P_i \cap T_i|}{\sum_i |P_i|}$$

Where P_i is the set consisting of the most classes predicted for each test example i, and all of its ancestor classes; T_i is the set consisting of the true most specific class(es) of test example i, and all ancestor classes.

2.3.2 Hierarchical Recall

Similar to hierarchical precision, hierarchical recall (Rec_h) measures the total capture of correct predictions. It is expressed as:

$$Rec_h = \frac{\sum_i |P_i \cap T_i|}{\sum_i |T_i|}$$

2.3.3 Hierarchical F1 Score

The hierarchical F-1 score (F_h) combines both hierarchical precision and recall (using a harmonic mean) to provide a single measure of model performance. It is computed as:

$$F_h = \frac{2 * Pr_h * Rec_h}{Pr_h + Rec_h}$$

This is also the official evaluation metric used to rank performance in *Subtask 1* and *Subtask 2a*.

3 Analysis

We benchmark the performance of our progressively fine-tuned model, and its intermediates, on the Subtask 1 development set. To further explore multilingual capabilities across post-training steps, we create 16 translated versions[5] of the held-out data encompassing a wide variety of languages.

[5]Translation was performed by the Google Translate API: `cloud.google.com/translate`

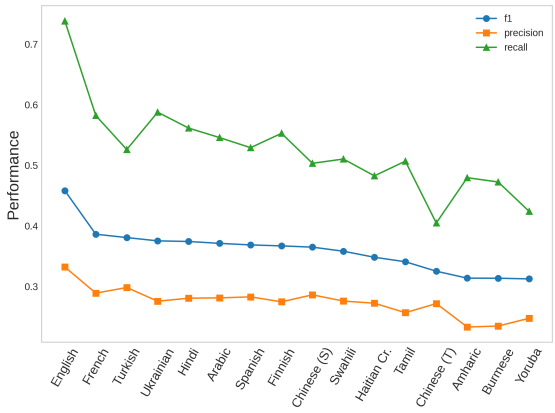


Figure 2: **Primed model (intermediate)**: Recall remains higher than precision in the intermediate model, generally indicating over-prediction. We also find that the relative performance of language varieties shift substantially.

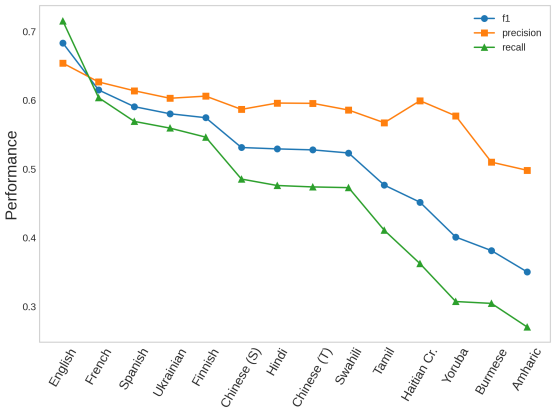


Figure 3: **Final Model**: The structure-tuned model exhibits the highest performance for most languages, included English.

3.1 Zero-Shot

We evaluate the capabilities of three popular out-of-the-box LLMs: OpenAI's GPT 3.5 Turbo, Meta's Llama 2 70B Chat model, and Mistral AI's Mixtral 8x7B instruct mixture of experts (MoE) model. Despite some variation in training data and architecture (see table 2), our tests reveal a consistent bias towards more-common languages (or those closely related to common languages). Furthermore, we find that multilingual capabilities do extend, at least in part, to the propaganda detection task.

3.2 Intermediate Model

After the first fine-tuning step (see section 2.2.1), we again evaluate how the 'meaning-primed' LLM performs in a multilingual setting in fig. 2. Despite English-only fine-tuning data, we find within-language performance improvements in nearly all settings. Our results also indicate that this step also improved some languages more than others, however these shifts do not have any clear syntactic, orthographic, or morphological basis.

3.3 Final Model

After the structural fine-tuning step described in section 2.2.2 , hierarchical F1, precision, and recall demonstrate further gains (see fig. 3). Again, despite English-only data, most languages[6] outperform zero-shot and intermediate counterparts. This is the version of the model which produced our official submissions for *Subtask 1*.

3.4 Multilingual Gains

In addition to producing the highest overall scores (likely a consequence of English-dominant pre-training and fine-tuning data), English also demonstrated the highest gain from additional data, as summarized in fig. 4. While both the priming and structural adaptation phases contributed positively, our results show that the latter was generally more impactful. We hypothesize that labeling differences across related subtask data prevented further performance increases between zero-shot and intermediate evaluation contexts. However, the minor modifications to the evaluation function which allowed for non-exact Python syntax and technique capitalization in the intermediate step would likely only serve to boost reported metrics.

4 Conclusion

Our work highlights the challenges inherent in adapting language models to tasks where relevant information deviates in format and/or linguistic scope from that of the desired output. Our results indicate that progressive fine-tuning offers a promising method for bridging this gap. By tailoring a standard LLM to effectively identify persuasion techniques within multilingual memes, we demonstrate the potential for decoupling syntactic requirements from task-specific 'understanding'. Although monolingual in post-training, this method yielded performance gains across all evaluated lan-

[6]Due to orthographic complications, we were unable to perform a final analysis on Arabic and Turkish.

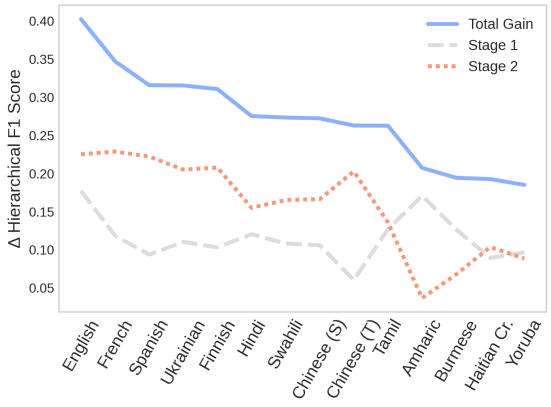


Figure 4: **Relative Performance**: English produced the highest overall increase between zero-shot and final performance, with the highest delta (in percentage points) coming from Stage 2 of the fine-tuning process.

guages compared to zero-shot settings, implying similar capabilities across a wide variety of use cases.

Nevertheless, this work prompts further questions regarding the interplay between pre-training corpora, post-training regimes, and the nature of evaluation data. Our results also call for further work in understanding how the data integration process impacts downstream performance– specifically in comparing our progressive fine-tuning approach to more common single-stage methods. Crucially, our findings reinforce the urgent need to investigate and mitigate biases in LLMs (Lai et al., 2023; Navigli et al., 2023) that impact their performance across varied language communities and use cases.

5 Acknowledgments

We thank Simon Todd for his input and suggestions. We also acknowledge the task organizers for designing such a topical and valuable challenge.

References

Julia R DeCook. 2018. Memes and symbolic violence:# proudboys and the use of memes for propaganda and the construction of collective identity. *Learning, Media and Technology*, 43(4):485–504.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Detecting Propaganda Techniques in Memes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6603–6617, Online. Association for Computational Linguistics.

Mohamed K. Elhadad, Kin Fun Li, and Fayez Gebali. 2020. Detecting Misleading Information on COVID-19. *IEEE access: practical innovations, open solutions*, 8:165201–165215.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv preprint arXiv:2304.05613*.

Jinfen Li, Zhihao Ye, and Lu Xiao. 2019. Detection of Propaganda Using Logistic Regression. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Scot Macdonald. 2006. *Propaganda and Information Warfare in the Twenty-First Century: Altered images and deception operations*. Routledge.

Roberto Navigli, Simone Conia, and Björn Ross. 2023. Biases in large language models: Origins, inventory and discussion. *ACM Journal of Data and Information Quality*.

Vitaliia-Anna Oliinyk, Victoria Vysotska, Yevhen Burov, Khrystyna Mykich, and Vítor Basto Fernandes. 2020. Propaganda Detection in Text Data Based on NLP and Machine Learning. In *MoMLeT+DS*.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data mining and knowledge discovery*, 22:31–72.

Cassian Sparkes-Vian. 2019. Digital propaganda: The tyranny of ignorance. *Critical sociology*, 45(3):393–409.

Taha ValizadehAslani, Yiwen Shi, Jing Wang, Ping Ren, Yi Zhang, Meng Hu, Liang Zhao, and Hualou Liang. 2022. Two-Stage Fine-Tuning: A Novel Strategy for Learning Class-Imbalanced Data. ArXiv:2207.10858 [cs].

Haoran Xu, Seth Ebner, Mahsa Yarmohammadi, Aaron Steven White, Benjamin Van Durme, and Kenton Murray. 2021. Gradual Fine-Tuning for Low-Resource Domain Adaptation. ArXiv:2103.02205 [cs].

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*.

A Appendix

```
[{...}, {
      "id": "125",
      "text": "I HATE TRUMP\n\nMOST TERRORIST DO",
      "labels": [
        "Loaded Language",
        "Name calling/Labeling"
    ],
    "link": "https://..."
    }, {...}...]
```

Listing 1: Pre-formatting .json snippet

```
{...}, {
    "messages": [
        {"role": "system", "content": ""},
        {"role": "user", "content": "I HATE TRUMP\n\
     nMOST TERRORIST DO"},
        {"role": "assistant", "content": "["Loaded
     Language","Name calling/Labeling"]"}
    ]
}, {...}
```

Listing 2: Post-formatting .jsonl snippet

GPT 3.5-Turbo zero-shot prompt

System Prompt: Respond only with a python list, nothing more. Identify which, if any, of the following propoganda labels apply to the given meme: ['Name Calling','Doubt','Smears','Reductio ad Hitlerum','Bandwagon','Glittering Generalities','Exaggeration','Loaded Language','Flag Waving','Appeal to Fear','Slogans','Repetition','Intentional Vagueness','Straw Man','Red Herring','Whataboutism','Causal Oversimplification','Black & White Fallacy','Thought Terminating Cliché']

Input: <MEME TEXT>

Mixtral 8x7b zero-shot prompt

Input: Respond only with a python list, nothing more. Identify which, if any, of the following propaganda labels apply to the given meme: ['Name Calling','Doubt','Smears','Reductio ad Hitlerum','Bandwagon','Glittering Generalities','Exaggeration','Loaded Language','Flag Waving','Appeal to Fear','Slogans','Repetition','Intentional Vagueness','Straw Man','Red Herring','Whataboutism','Causal Oversimplification','Black & White Fallacy','Thought Terminating Cliché']. Meme: <MEME TEXT>

Llama 2 70b zero-shot prompt

Input: Respond only with a python list, nothing more. Identify which, if any, of the following propaganda labels apply to the given meme: ['Name Calling','Doubt','Smears','Reductio ad Hitlerum','Bandwagon','Glittering Generalities','Exaggeration','Loaded Language','Flag Waving','Appeal to Fear','Slogans','Repetition','Intentional Vagueness','Straw Man','Red Herring','Whataboutism','Causal Oversimplification','Black & White Fallacy','Thought Terminating Cliché']. Meme: <MEME TEXT>

NLU-STR at SemEval-2024 Task 1: Generative-based Augmentation and Encoder-based Scoring for Semantic Textual Relatedness

Sanad Malaysha, Mustafa Jarrar, Mohammed Khalilia
Birzeit University, Palestine
{smalaysha, mjarrar, mkhalilia}@birzeit.edu

Abstract

Semantic textual relatedness is a broader concept of semantic similarity. It measures the extent to which two chunks of text convey similar meaning or topics, or share related concepts or contexts. This notion of relatedness can be applied in various applications, such as document clustering and summarizing. SemRel-2024, a shared task in SemEval-2024, aims at reducing the gap in the semantic relatedness task by providing datasets for fourteen languages and dialects including Arabic. This paper reports on our participation in Track A (Algerian and Moroccan dialects) and Track B (Modern Standard Arabic). A BERT-based model is augmented and fine-tuned for regression scoring in supervised track (A), while BERT-based cosine similarity is employed for unsupervised track (B). Our system ranked 1st in SemRel-2024 for MSA with a Spearman correlation score of 0.49. We ranked 5th for Moroccan and 12th for Algerian with scores of 0.83 and 0.53, respectively.

1 Introduction

The literature commonly examines semantic similarity, where the focus is on whether two linguistic units (words, phrases, sentences, etc.) share similar meanings (Bentivogli et al., 2016). However, semantic textual relatedness (STR) is less explored due to its complexity and the scarcity of datasets (Abdalla et al., 2023; Darwish et al., 2021). While the former task checks for the presence of similar meaning or paraphrase, STR takes a more comprehensive approach, evaluating relatedness across multiple dimensions, spanning topical similarity, conceptual overlap, contextual coherence, pragmatic connection, themes, scopes, ideas, stylistic conditions, ontological relations, entailment, temporal relation, as well as semantic similarity itself (Miller and Charles, 1991; Halliday and Hasan, 2014; Jarrar, 2021, 2011). For example, consider the two sentences (*The Earth orbits the sun at a speed of ~110,000 km/h.*) and (*Earth rotates at ~1670 km/h around its axis.*). They hold semantic relatedness through the shared topic of Earth's speeds. In contrast, both sentences are not semantically similar as they possess distinct meanings. This illustrates the broader range of STR as described by Abdalla et al. (2023), which ranges from highly relevant sentences, expressing the same idea with different wording, to entirely unrelated sentences, discussing unrelated topics.

Semantic relatedness has proven to be useful in evaluating sentence representations generated by language models (Asaadi et al., 2019), in addition to question answering (Tsatsaronis et al., 2014), machine translation (Mi and Xie, 2024), plagiarism detection (Sabir et al., 2019), word-sense disambiguation (Al-Hajj and Jarrar, 2021a; Malaysha et al., 2023), among others. Exploring the relatedness and similar tasks in languages other than English is hindered by the lack of data (Jarrar et al., 2023b; Al-Hajj and Jarrar, 2021b). The SemRel-2024 shared task (Ousidhoum et al., 2024a) provided datasets in fourteen languages and offered three tracks. In the supervised track (A), training and testing are performed on the same language. In the unsupervised track (B), the use of labeled data for training is prohibited; and in the cross-lingual track (C), testing is conducted on a different language than the one used for training.

This paper presents our contribution to track A and track B. In track A, we fine-tuned BERT models using the Algerian and Moroccan sentence pairs to produce similarity scores. To enrich the data, we augmented the SemRel-2024 dataset (Ousidhoum et al., 2024a) by generating additional sentence pairs from Google Gemini [1], a generative model, using a predefined prompt template. These generated pairs imitated the style and meaning of the existing pairs, and we assigned them

[1] https://gemini.google.com/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 894–901

scores corresponding to the originals. We used the same datasets provided by the Shared Task in addition to a ~760 augmented Moroccan pairs to fine-tune BERT models, AraBERTv2 (Antoun et al., 2020) and ArBERTv2 (Abdul-Mageed et al., 2021), which resulted in a performance enhancement of 0.05 points. In track B, as training on labeled data is not allowed, we used cosine similarity using average pooling embedding (Zhao et al., 2022) on top of each model. Our approaches achieved Spearman scores (Tsatsaronis et al., 2014) of 0.49 for MSA (ranked first), 0.83 for Moroccan (ranked fifth), and 0.53 for Algerian (ranked twelfth).

2 Related Work

Semantic textual relatedness (STR) has proven to be a valuable task in numerous NLP applications, including the evaluation of LLMs (Asaadi et al., 2019; Naseem et al., 2021). Determining the degree of relatedness in STR, however, remains a challenging task in computational semantics. That is because STR encompasses a broader range of commonalities beyond just meaning, including shared viewpoint, topic, and period, demanding a deeper understanding than semantic similarity alone (Asaadi et al., 2019; Abdalla et al., 2023). For example, consider reading these two sentences (*He heard the waves crashing gently*) and (*Making him feel calm and peaceful*). While humans easily recognize their strong relatedness and shared description of the same view (a beach scene), machines require advanced lexical and statistical methods to achieve the same level of understanding. STR techniques mainly come from four approaches: lexical similarity (Chen et al., 2018; Jarrar and Amayreh, 2019; Alhafi et al., 2019), semantic similarity (Hasan et al., 2020; Ghanem et al., 2023), deep learning (Zhang and Moldovan, 2019), and LLMs (Li et al., 2021).

Recently, Abdalla et al. (2023) introduced their STR-2022 dataset, which uses fine-grained scores ranging from 0 (least related) to 1 (completely related). Their dataset consists of 5,500 scored English sentence pairs. They framed the task as supervised regression, where they fine-tuned two language models, BERT-base (Kenton and Toutanova, 2019) and RoBERTa-base (Liu et al., 2019), and applied average pooling on top of the final embedding layer. Their testing of these models on the STR-2022 dataset yielded an average Spearman correlation of 0.82 for BERT-base and 0.83 for RoBERTa-base. On the other hand, their unsupervised experiments using Word2Vec (Mikolov et al., 2013) achieved a correlation score of 0.60, outperforming both BERT-base (0.58) and RoBERTa-base (0.48) by 0.02 and 0.12 points, respectively.

Asaadi et al. (2019) created the Bi-gram Semantic Relatedness Dataset (BiRD) for examining semantic composition. To avoid inconsistencies and biases from traditional 1-5 rating scales, they employed fine-grained scoring of bi-gram pairs (0-1) using the best-worst scaling (BWS) annotation technique (Kiritchenko and Mohammad, 2017). The dataset consists of 3,345 scored English term pairs. They utilised three models to generate word representations: GloVe (Pennington et al., 2014), FastText (Grave et al., 2018), and a word-context co-occurrence matrix (Turney et al., 2011). To calculate relatedness scores between pairs, they employed cosine similarity between the generated addition-pooled vectors. The FastText model achieved the highest performance with a Pearson correlation of 0.60.

The semantic relatedness between noun-pairs was studied using contextual similarity by Miller and Charles (1991). They attempted to understand distinctions between nouns in contextual discourse and how the similarity can be broader than just the meaning. Additional ideas could rely on extracting named entities (Liqreina et al., 2023; Jarrar et al., 2022) to measure the relatedness (Ghosh et al., 2023). However, the task evolved, leading to the creation of the up-to-date dataset presented by the SemRel-2024 shared task (Ousidhoum et al., 2024b). Their dataset annotation scores are at the level of sentence pairs. They shared baseline results for fourteen languages and dialects using Spearman correlation score. Since our focus is on Arabic, we have chosen its results to show. For example, their baseline is 0.42 for MSA in track B using multilingual BERT (mBERT) (Kenton and Toutanova, 2019), 0.60 for Algerian and 0.77 for Moroccan in track A using Label Agnostic BERT Sentence embeddings (LaBSE) (Feng et al., 2022). Specifically, their Algerian Arabic dataset offers 1,261 training and 583 test instances, Moroccan Arabic dataset includes 924 training and 425 test instances, and MSA Arabic dataset has 595 instances for testing.

Many efforts have been made to understand Arabic dialects, such as dialect identification, intent detection, and morphological annotations (Haff et al., 2022; Nayouf et al., 2023; Jarrar et al., 2023c, 2017, 2023a), but none studied STR between dialects.

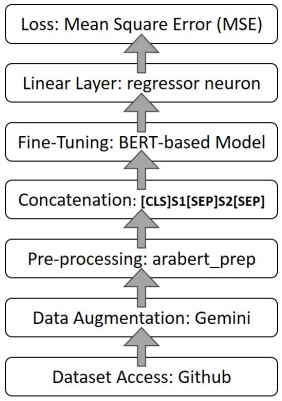


Figure 1: BERT-based Supervised Architecture (A).

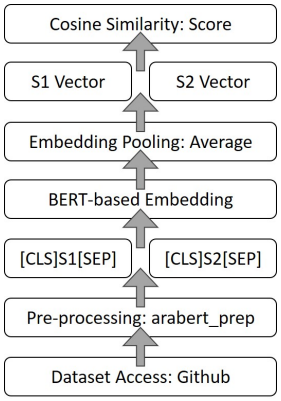


Figure 2: BERT-based Unsupervised Architecture (B).

3 System Overview

This section presents the techniques, datasets, and the augmentation we employed in tracks A and B.

3.1 Supervised Track (A)

Since the datasets use continuous scoring values, we tackled STR as a regression problem. We fine-tuned BERT with Mean Squared Error (MSE) objective. The model uses a regressor output layer, represented by a single neuron to predict the scores of the sentence-pairs. The data was pre-processed using the technique presented in (Antoun et al., 2020) to achieve standardized word forms. Before supplying the sentence pairs to the model, each was concatenated using the special tokens of the model input in this format:`[CLS]Sentence1[SEP]Sentence2[SEP]`. Figure 1 depicts our method architecture for the supervised track (A). Since we focused on the Algerian and Moroccan dialects in this track, we investigated various model parameters including learning rates, number of epochs, and pre-trained models to understand which model is better suited for each dialect. We found that both models, AraBERTv2 [2] and ArBERTv2 [3], best fits the Moroccan dialect more than Algerian. Nonetheless, we used same models for the Algerian dataset.

3.2 Unsupervised Track (B)

The STR using MSA is covered in track B (unsupervised learning), where training (or fine-tuning) on labeled data is not permitted. We employed cosine similarity (Reimers and Gurevych, 2019) as an unsupervised technique to calculate the sentence-pair scores. Figure 2 illustrates our architecture. We conducted initial experiments using the same aforementioned models, ArBERTv2 and AraBERTv2, for generating sentence representations. Various pooling options (CLS, average, max, and min) (Zhao et al., 2022) were applied on the final embedding layer in each (frozen) model, and found that AraBERTv2 with average-pooling is better suited for MSA in this track. The same data pre-processing used in track A is applied in B.

3.3 Datasets

The datasets provided by the SemRel-2024 shared task cover fourteen languages and dialects. In the paper, we used three Arabic datasets (Algerian, Moroccan, and MSA). Table 1 presents their data splits, including train, development, and testing. MSA has no labeled train data as it is included in Track B. However, for the other two dialects, we employed BERT-based models, that requires large train data (Bevilacqua et al., 2021).

	MSA	Algerian	Moroccan
Train			
Original	–	1,261	924
Augments	–	–	757
Total	–	1,261	1,681
Dev.	32	97	70
Test	595	583	425

Table 1: The original and augmented datasets splits.

Different methods can be used for data augmentation, such as back-translation (Lin and Giambi, 2021) and generative models (Saidi et al., 2022). The back-translation technique was tested by (Malaysha et al., 2023) and showed minor improvement in performance. The availability of high-

[2] https://github.com/aub-mind/arabert
[3] https://huggingface.co/UBC-NLP/ARBERTv2

Original Sentence 1	Original Sentence 2	Score
كورونا..12 تقاسو بالفيروس فجهة العيون الساقية الحمراء	كورونا: 99 تقاسو بالفيروس فجهة العيون الساقية الحمراء	0.79
Original Sentence 1	**Augmented Sentence 2**	
كورونا..12 تقاسو بالفيروس فجهة العيون الساقية الحمراء	باقة تاخدات فالفيروس فلعيون الساقية الحمراء. 99 حالة بزاف	0.79
Augmented Sentence 1	**Original Sentence 2**	
فـ جهة العيون الساقية الحمراء، 12 تصاوبو بكورونا	كورونا: 99 تقاسو بالفيروس فجهة العيون الساقية الحمراء	0.79

Figure 3: Example of the augmented sentence-pairs.

quality generative models, such as ChatGPT [4] and Google Gemini, encouraged us to employ them in automatic augmentation. We employed in-context learning (Min et al., 2022) by prompting both models with the request depicted in Figure 4.

Prompt

Augment the following Arabic sentence using Moroccan dialect. Please generate Moroccan sentence similar in meaning to the one I provide you, and use average number of words close to the length of the provided sentence. You have to format the augmented sentence between pair of box brackets []. Do not add any explanations, I just need the reply same the format I provided without any additional texts or confirmations. I will repeat this request hundreds of times using different sentences, so do not change the format of your reply. The sentence is in Moroccan dialect:
كورونا..12 تقاسو بالفيروس فجهة العيون الساقية الحمراء

Reply: [فـ جهة العيون الساقية الحمراء، 12 تصاوبو بكورونا]

Figure 4: The prompt template employed for Gemini.

The initial manual reviews and tests for twenty prompts of Moroccan and Algerian sentences showed that both models are weak in Algerian comprehension. ChatGPT is also weak in the Moroccan, while Gemini demonstrated a high understanding of the Moroccan. Therefore, we decided to employ Gemini to augment the Moroccan train split. From every sentence-pair, we took each sentence and prompted it using the template in Figure 4. We mapped the augmented (new) sentence from the model with the other sentence in the same pair using the same score of the pair, as illustrated in Figure 3. By manually reviewing all the model replies, we found cases that were not valid (wrong content), and accordingly, we defined filters to exclude the not applicable data per the following rules:

- The model admits in the reply that it is just a language model and cannot fulfill the request. The model reply in such case has common format to rely on for the filter comparison.
- The case when the reply goes far from the original meaning. This option is achieved by manually reviewing the paraphrased contents.
- When the model rejects augmentation because the requested sentence contains information that breaks the model policy, i.e., talking about public figures or sensitive discussions. Similar to first rule, it has common reply format to automatically compare with.

Finally, after filtering the invalid augmentations, we reached 757 accepted sentences which we added to the Moroccan training set (See Table 1), reaching a total of 1,681 instances.

4 Experimental Setup

Our experiments fine-tuned two language models for Algerian and Moroccan, where we used the following pre-trained models: maubmindlab/bert-base-arabertv02 (Antoun et al., 2020) and UBC-NLP/ARBERTv2 (Abdul-Mageed et al., 2021). We employed the training data provided by the shared task, in addition to the data generated by our augmentation technique, when applied. The development data is excluded from either training or testing in the official evaluation phase, and testing is done on the shared task test set (See Table 1). The data pairs were concatenated using special tokens (`[CLS]` and `[SEP]`), as depicted in Figure 1, and digested by the models. The fine-tuning was done as a regression task using one neuron in the output layer, optimized using MSE as the loss function, and we used R-squared (Miles, 2005) to measure the improvement. The final hyper-parameters in the fine-tuning process were: 10 epochs for training, 4 epochs for early stopping, a batch size of 16,

[4] https://chat.openai.com/

Development Phase	Track A			Track B
	Algerian	Moroccan	Augmented Moroccan	MSA
ArBERTv2	0.55	0.82	0.88↑	0.42
AraBERTv2	0.69	0.84	0.79↓	0.58

Table 2: Our results on the development phase (i.e., on development split).

TEST Phase	Track A			Track B
	Algerian	Moroccan	Augmented Moroccan	MSA
Baseline (Ousidhoum et al., 2024a)	0.60	0.77	0.77	0.42
ArBERTv2	0.42↓	0.78↑	**0.83**↑	0.34↓
AraBERTv2	**0.53**↓	0.79↑	0.77↑	**0.49**↑

Table 3: The evaluation results on the test data. Our official ranked scores are in bold.

512 is the maximum sequence length, a learning rate of $2e^{-5}$, 50 evaluation steps, a seed of 42, and train ($\pm$ augmented data) split.

In the experiments of B track for the MSA, no supervised fine-tuning is needed. Therefore, we neither used labeled data nor augmentation. We employed average-pooling on the embeddings of the sentence tokens from the final layer in each model. Then, we calculated the cosine similarity between the average embeddings of the sentences in each pair. This was done to estimate the fine-grained scores for the test (or development) data provided by the shared task. The shared task considers Spearman correlation score to evaluate the submitted predictions against their ground truth.

5 Results

Our approaches have achieved competitive ranks in the SemRel-2024 shared task. The official results of the tracks we participated in, as well as the baselines that were introduced by Ousidhoum et al. (2024a), are shown in Table 3. Additionally, our results on the development data are presented in Table 2. In the test evaluation, we ranked first in Track B for the MSA, with a Spearman correlation score of 0.49 using the AraBERTv2 model, outperforming the baseline by 0.07 points. However, ArBERTv2 did not perform well in Track B for MSA on both test and development splits. In contrast, ArBERTv2 achieved a high score in Track A for the Moroccan dialect when fine-tuned on both the train split and augmentation data, outperforming the baseline by 0.06 points on test split, ranking 5th among the submitted systems. Nonetheless, neither of the models, ArBERTv2 or AraBERTv2, surpassed the baseline for the Algerian dialect in Track A, where our rank is 12. Similarly, both models achieved low performance on the Algerian development split. It is possible that if we were able to augment the Algerian data as well, it could have performed better, similar to the improvement achieved in the Moroccan dataset. It is worth noting that AraBERTv2 outperformed both the baseline and ArBERTv2 on the original training data of the Moroccan dataset. However, its performance degraded on both test and development splits once the augmentation was included in the fine-tuning, unlike what happened with the ArBERTv2 model, on both splits. This could be due to the nature of the data utilized in the pre-training phase of the model. Due to the anisotropy problem (Baggetto and Fresno, 2022) inherent in BERT-based pre-trained models, we noted that computing cosine similarity directly between sentence representations is insufficient for discerning relatedness.

6 Conclusion

We presented our contributions to the SemRel-2024 shared task. We targeted three Arabic dialects covered by the shared task datasets, including MSA, Algerian, and Moroccan. Our approaches employed supervised and unsupervised techniques using commonly known language models, namely ArBERT and AraBERT. We augmented the training data using generative models, which enhanced the models' performance. Our system ranked first (MSA), fifth (Moroccan), and twelfth (Algerian) across the different tracks. We plan to augment additional data of Moroccan and Algerian using other models than what we used in this work. We will use the augmentations to experiment with both Arabic mono-dialect and cross-dialect fine-tuning.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2023. What makes sentences semantically related? A textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 782–796. Association for Computational Linguistics.

Muhammad Abdul-Mageed, AbdelRahim A. Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: deep bidirectional transformers for arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7088–7105. Association for Computational Linguistics.

Moustafa Al-Hajj and Mustafa Jarrar. 2021a. Arabgloss-bert: Fine-tuning bert on context-gloss pairs for wsd. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 40–48, Online. INCOMA Ltd.

Moustafa Al-Hajj and Mustafa Jarrar. 2021b. Lu-bzu at semeval-2021 task 2: Word2vec and lemma2vec performance in arabic word-in-context disambiguation. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 748–755, Online. Association for Computational Linguistics.

Diana Alhafi, Anton Deik, and Mustafa Jarrar. 2019. Usability evaluation of lexicographic e-services. In *The 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7. IEE.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15.

Shima Asaadi, Saif M. Mohammad, and Svetlana Kiritchenko. 2019. Big bird: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 505–516. Association for Computational Linguistics.

Alejandro Fuster Baggetto and Víctor Fresno. 2022. Is anisotropy really the cause of BERT embeddings not being semantic? In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4271–4281. Association for Computational Linguistics.

Luisa Bentivogli, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. 2016. SICK through the semeval glasses. lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Lang. Resour. Evaluation*, 50(1):95–124.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, Roberto Navigli, et al. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338.

Zugang Chen, Jia Song, and Yaping Yang. 2018. An approach to measuring semantic relatedness of geographic terminologies using a thesaurus and lexical database sources. *ISPRS Int. J. Geo Inf.*, 7(3):98.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab worlds. *Commun. ACM*, 64(4):72–81.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics.

Sana Ghanem, Mustafa Jarrar, Radi Jarrar, and Ibrahim Bounhas. 2023. A benchmark and scoring algorithm for enriching arabic synonyms. In *Proceedings of the 12th International Global Wordnet Conference (GWC2023)*, pages 215–222. Global Wordnet Association.

Mirna El Ghosh, Nicolas Delestre, Jean-Philippe Kotowicz, Cecilia Zanni-Merk, and Habib Abdulrab. 2023. Reltopic: A graph-based semantic relatedness measure in topic ontologies and its applicability for topic labeling of old press articles. *Semantic Web*, 14(2):293–321.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomás Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a levantine corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. 2014. *Cohesion in english*. 9. Routledge.

Ali Muttaleb Hasan, Noorhuzaimi Mohd Noor, Taha H Rassem, and Ahmed Muttaleb Hasan. 2020. Knowledge-based semantic relatedness measure using semantic features. *International Journal*, 9(2).

Mustafa Jarrar. 2011. Building a formal arabic ontology (invited paper). In *Proceedings of the Experts Meeting on Arabic Ontologies and Semantic Networks*. ALECSO, Arab League.

Mustafa Jarrar. 2021. The arabic ontology - an arabic wordnet with ontologically clean content. *Applied Ontology Journal*, 16(1):1–26.

Mustafa Jarrar and Hamzeh Amayreh. 2019. An arabic-multilingual database with a lexicographic search engine. In *The 24th International Conference on Applications of Natural Language to Information Systems (NLDB 2019)*, volume 11608 of *LNCS*, pages 234–246. Springer.

Mustafa Jarrar, Ahmet Birim, Mohammed Khalilia, Mustafa Erden, and Sana Ghanem. 2023a. Arbanking77: Intent detection neural model and a new dataset in modern and dialectical arabic. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*, pages 276–287. ACL.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: An annotated corpus for the palestinian arabic dialect. *Journal Language Resources and Evaluation*, 51(3):745–775.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested arabic named entity corpus and recognition using bert. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Mustafa Jarrar, Sanad Malaysha, Tymaa Hammouda, and Mohammad Khalilia. 2023b. Salma: Arabic sense-annotated corpus and wsd benchmarks. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*, pages 359–369. ACL.

Mustafa Jarrar, Fadi Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2023c. Lisan: Yemeni, irqi, libyan, and sudanese arabic dialect copora with morphological annotations. In *The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Svetlana Kiritchenko and Saif M. Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 465–470. Association for Computational Linguistics.

Xiaotao Li, Shujuan You, and Wai Chen. 2021. Enhancing accuracy of semantic relatedness measurement by word single-meaning embeddings. *IEEE Access*, 9:117424–117433.

Guan-Ting Lin and Manuel Giambi. 2021. Context-gloss augmentation for improving word sense disambiguation. *arXiv preprint arXiv:2110.07174*, abs/2110.07174.

Haneen Liqreina, Mustafa Jarrar, Mohammed Khalilia, Ahmed Oumar El-Shangiti, and Muhammad Abdul-Mageed. 2023. Arabic fine-grained entity recognition. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*, pages 310–323. ACL.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Sanad Malaysha, Mustafa Jarrar, and Mohammed Khalilia. 2023. Context-gloss augmentation for improving arabic target sense verification. In *Proceedings of the 12th Global Wordnet Conference, GWC 2023, University of the Basque Country, Donostia - San Sebastian, Basque Country, Spain, 23 - 27 January 2023*, pages 254–262. Global Wordnet Association.

Chenggang Mi and Shaoliang Xie. 2024. Language relatedness evaluation for multilingual neural machine translation. *Neurocomputing*, 570:127115.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Jeremy Miles. 2005. R-squared, adjusted r-squared. *Encyclopedia of statistics in behavioral science*.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics.

Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. 2021. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 20(5):74:1–74:35.

Amal Nayouf, Mustafa Jarrar, Fadi zaraket, Tymaa Hammouda, and Mohamad-Bassam Kurdy. 2023. Nâbra: Syrian arabic dialects with morphological annotations. In *Proceedings of the 1st Arabic Natural Language Processing Conference (ArabicNLP), Part of the EMNLP 2023*, pages 12–23. ACL.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Ahmed Sabir, Francesc Moreno, and Lluís Padró. 2019. Semantic relatedness based re-ranker for text spotting. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3449–3455. Association for Computational Linguistics.

Rakia Saidi, Fethi Jarray, Jeongwoo Jay Kang, and Didier Schwab. 2022. GPT-2 contextual data augmentation for word sense disambiguation. In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation, PACLIC 2022, Manila, Philippines, October 20-22, 2022*, pages 455–462. De La Salle University.

George Tsatsaronis, Iraklis Varlamis, and Michalis Vazirgiannis. 2014. Text relatedness based on a word thesaurus. *CoRR*, abs/1401.5699.

Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 680–690. ACL.

Linrui Zhang and Dan Moldovan. 2019. Multi-task learning for semantic relatedness and textual entailment. *Journal of Software Engineering and Applications*, 12(6):199–214.

Shuai Zhao, Tianyu Zhang, Man Hu, Wen Chang, and Fucheng You. 2022. AP-BERT: enhanced pre-trained model through average pooling. *Appl. Intell.*, 52(14):15929–15937.

scaLAR SemEval-2024 Task 1: Semantic Textual Relatednes for English

M Hemanth Kumar and **Anand Kumar M**
Artificial Intelligence, Department of Information Technology
National Institute of Technology Karnataka
Surathkal, India
mogilipalemhemanthkumar@gmail.com,m_anandkumar@nitk.edu.in

Abstract

This study investigates Semantic Textual Related- ness (STR) within Natural Language Processing (NLP) through experiments conducted on a dataset from the SemEval-2024 STR task. The dataset comprises train instances with three features (PairID, Text, and Score) and test instances with two features (PairID and Text), where sentence pairs are separated by '/n' in the Text column. Using BERT(sentence transformers pipeline), we explore two approaches: one with fine-tuning (Track A: Supervised) and another without fine-tuning (Track B: UnSupervised). Fine-tuning the BERT pipeline yielded a Spearman correlation coefficient of 0.803, while without fine-tuning, a coefficient of 0.693 was attained using cosine similarity. The study concludes by emphasizing the significance of STR in NLP tasks, highlighting the role of pre-trained language models like BERT and Sentence Transformers in enhancing semantic relatedness assessments.

1 Introduction

Semantic Textual Relatedness (STR) is a crucial concept in natural language processing (NLP), focusing on determining the degree of similarity between linguistic units like words or sentences based on their meaning. This measure plays a vital role in evaluating the effectiveness of Large Language Models (LLMs) and aids in various NLP tasks. At its core, STR delves into understanding the closeness in meaning between two pieces of text. It examines different dimensions of relatedness, including sharing the same viewpoint, originating from the same context, or complementing each other's content. For instance, if two sentences convey similar ideas through paraphrasing or entailment, they might be considered semantically similar. However, relatedness en- compasses all possible commonalities between them. In NLP, researchers and practitioners leverage STR to enhance textual coherence, refine narrative structures, and tackle diverse language understanding challenges. By quantifying semantic relatedness, NLP systems can better comprehend and generate human-like responses, ultimately advancing the capabilities of language models.

The concept of semantic relatedness between language units has been recognized as foundational in understanding meaning. The automatic determination of relatedness has found numerous applications, including the evaluation of sentence representation methods, question answering, and summarization. Semantically similar sentences are those that exhibit either a paraphrasal or entailment relationship. In contrast, relatedness encompasses a broader spectrum of commonalities between two sentences. This includes considerations such as whether they pertain to the same topic, convey the same perspective, emerge from the same temporal context, or if one sentence elaborates on or logically follows from the other. Despite the significance of relatedness, much of the prior work in natural language processing has predominantly fo- cused on semantic similarity, particularly within the context of English.

We Explored SBERT. Sentence-BERT(Reimers and Gurevych, 2019) builds upon the architecture of BERT (Bidirectional Encoder Representations from Transformers)(Devlin et al., 2018), leveraging transformer-based models to encode contextual information from input sentences. Unlike BERT, which focuses on token-level representations, Sentence-BERT aims to generate fixed-size representations for entire sentences. To achieve this, Sentence-BERT employs siamese or triplet network architectures, which are trained on sentence pairs or triplets with similar or dissimilar semantic meanings. Through contrastive loss functions, Sentence-BERT learns to map semantically similar sentences closer together in the embedding space while pushing dissimilar sentences farther

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 902–906

apart.

2 Background

The exploration of semantic relatedness in language finds its roots in seminal works by (Halliday and Hasan, 1976) and (Miller and Charles, 1991), which laid early foundations for understanding the subtleties of meaning in text. Initially, these efforts primarily focused on semantic similarity, assessing the likeness between linguistic units through techniques like paraphrasing or entailment. However, as research progressed, scholars began recognizing the necessity of considering a broader array of connections between text segments, thereby giving rise to the concept of semantic relatedness.

Semantic similarity denotes the extent of resemblance in meaning between two linguistic units, while semantic relatedness encompasses a wider spectrum of connections, encompassing elements such as topical relevance, viewpoint alignment, temporal coherence, and logical sequence. While semantic similarity often relies on paraphrasing or entailment, relatedness factors in various nuances contributing to the overall coherence and cohesion of text.

Traditional methodologies for measuring semantic relatedness relied on lexical and syntactic features, including word overlap, syntactic parse trees, and semantic networks. However, the emergence of deep learning techniques has ushered in a paradigm shift towards leveraging neural embeddings and transformer-based models to capture richer semantic representations. These modern approaches have demonstrated superior performance across various Semantic Textual Relatedness (STR) tasks, such as semantic similarity estimation and semantic textual entailment.

In the field of natural language processing (NLP), assessing the relatedness between pairs of sentences is a fundamental task. The paper(Hany et al., 2023) addresses this challenge by proposing an innovative approach that combines two key techniques. First, the authors leverage embedding similarity techniques, utilizing seven different transformers to generate sentence vectors. These vectors capture the semantic content of sentences, allowing for more accurate relatedness assessment. Second, a classical machine learning regressor is trained on these sentence vectors. By integrating these methods, the study achieved impressive results on the SICK dataset. Specifically, the mean square error is reduced to 0.0481, and high Pearson's and Spearman's correlations of 0.978 and 0.9696, respectively, demonstrate the effectiveness of this approach. Overall, this research highlights the potential of combining embedding similarity techniques with machine learning for improving relatedness score assessment and advancing NLP algorithms. The zero-shot text classification (0SHOT-TC) has garnered significant attention. This task involves detecting classes that the model has never encountered during training. The emergence of pre-trained language models has transformed 0SHOT-TC into a binary classification problem, akin to textual entailment. Specifically, the model learns whether there is an entailment-relatedness (yes/no) between a given sentence (premise) and each category (hypothesis). However, existing approaches struggle with fully expressing the category space using labels or label descriptions. In contrast, humans can effortlessly extend a set of words to describe the categories to be classified. To bridge this gap, the paper(Liu et al., 2023) introduces a novel method called Semantically Extended Textual Entailment (SETE). Inspired by human knowledge extension, SETE enriches category representations using a combination of static knowledge (e.g., expert knowledge, knowledge graphs) and dynamic knowledge (e.g., language models).

Early methods, such as Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF), fell short in capturing nuanced word meanings and context. To address this, the paper(Abdalla et al., 2021) surveys the evolution of semantic similarity techniques, categorizing them into knowledge-based, corpus-based, deep neural network-based, and hybrid approaches. By examining the strengths and limitations of each method, the survey provides a comprehensive overview for researchers navigating the complex landscape of semantic similarity research. Understanding the degree of semantic relatedness between two language units is fundamental. However, prior research has primarily focused on semantic similarity, a subset of relatedness, due to the scarcity of relatedness datasets. To address this gap, the authors(Chandrasekaran and Mago, 2021) introduce the Semantic Textual Relatedness (STR-2022) dataset, comprising 5,500 English sentence pairs manually annotated using a comparative annotation framework. Human intuition regarding sentence relatedness proves highly reliable, with a repeat annotation correlation of 0.84. The dataset not only

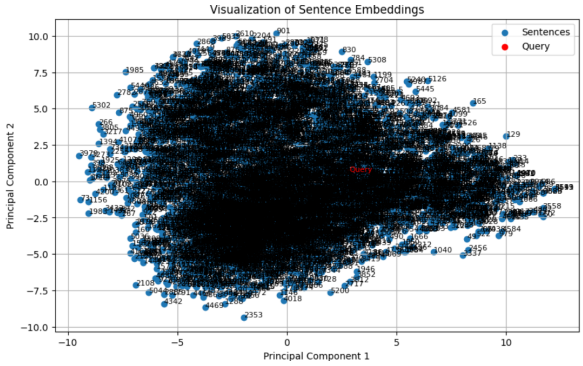


Figure 1: Visualization of Sentence Embeddings

facilitates exploration of what makes sentences semantically related but also serves as a valuable resource for evaluating automatic sentence representation methods and various downstream NLP tasks.

3 System Overview

A model in the Sentence-Transformers library is the bert-base-nli-mean-tokens collection. It is especially made for the purpose of Semantic Textual Similarity (STS). Sentences and paragraphs are mapped to a 768-dimensional dense vector space by this paradigm. The pre-training phase Bert-base-nli-mean-tokens are pre-trained using the conventional BERT architecture. This model is pretrained on the tasks of Modeling Masked Languages (MLM): Tokens in the training data are masked with a unique token [MASK] or randomly substituted with a small percentage, Bidirectional Contextualization: By analyzing both left and right context, BERT generates bidirectional contextualized embeddings as it learns to forecast masked tokens and Next Sentence Prediction (NSP): In the original text, BERT further forecasts if two sentences will come after one another. In learning sentence relationships, this aids the model. Fig -1 shows the visualization of Sentence embeddings from this model.

4 Experimental Setup

we considered the dataset(Ousidhoum et al., 2024a) from codalab SemEval-2024 STR task (Ousidhoum et al., 2024b). There are 5500 samples as train instances and three features namely PairID, Text ans Score . There are 250 samples as test instances and two features PairID and Text. In the Text column of the data, the pair of sentences are separated by '/n'. We conducted a couple of experiments

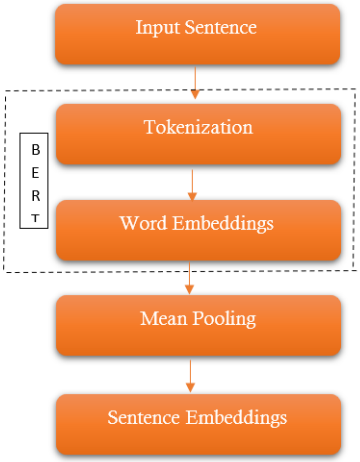


Figure 2: Methodology Used

for the Semantic Textual Relatedness on the above mentioned data using BERT. In our study, we utilized the bert-base-nli-mean-tokens model, a variant of the BERT architecture tailored for the Semantic Textual Similarity (STS) task. This model, an integral component of the Sentence Transformers (sbert) library, plays a pivotal role in assessing the semantic relatedness between pairs of text sentences. The designation 'base' signifies a medium-sized version of the BERT model, balancing computational efficiency with performance. Pre-trained on Natural Language Inference (NLI) tasks, the model captures intricate semantic relationships between text segments, essential for STS tasks. Additionally, employing mean pooling, it generates fixed-length sentence embeddings efficiently, providing a comprehensive representation of semantic content. Our utilization of bert-base-nli-mean-tokens in our research ensures robust and nuanced analysis of semantic similarity, contributing to advancements in natural language understanding and related fields.The fig- 2 shows the methodology we followed to extract sentence embeddings.

4.1 Without Fine Tuning

At First, we separated the two sentences by the delimited '/n'. we have the pair of sentences. Now, our task is to get the sentence embeddings for these sentences. We used Sentence Transformers pipeline, with the "bert-base-nli-mean-tokens" as the model.This particular model is based on the BERT (Bidirectional Encoder Representations from Transformers) architecture and is trained to generate sentence embeddings by taking the mean of the token embeddings.These sentence

embeddings are used to calculate Semantic Textual Relatedness by using custom defined Cosine Similarity function. The custom function computes the cosine similarity between two input vectors u and v(sentence Embeddings) . Utilizing NumPy's dot product and Euclidean norm functions, the function calculates the cosine similarity by dividing the dot product of the input vectors by the product of their Euclidean norms. Commonly employed in Natural Language Processing tasks, cosine similarity serves as a fundamental metric for comparing the semantic similarity between word embeddings or Sentence embeddings, facilitating various applications such as information retrieval and document clustering.
Every word in the text was mapped to a word embedding space using the model. The cosine distance between the two sentences was computed after the embeddings. Equation 1 illustrates how the cosine similarity between the two embedding vectors is computed.
$\cos(\theta) = \frac{A \cdot B}{\|A\|_2 \|B\|_2}$

The requested forecast for the two sentences under consideration was then given as the cosine similarity value.

4.2 Fine Tuning

Similar to the previous experiment, we separated the two sentences by the delimited '/n'. We used the same pipeline to generate sentence embeddings. But this time, instead of directly evaluating the performance of the model using cosine similarity. We first fine-tuned the model with the following parameters Table 1 and then we evaluated it's performance.

Table 1: Parameters Used

Parameter	Value
Batch size	16
Epochs	1
Loss	CosineSimilarityLoss
Optimizer	Adam

5 Results

On Google Colab, we put our approach into practice. Sentence Transformer was the library that we used. Pythorch7 (>=1.11.0) and Python 6 (>= 3.8) are required by the library. The Official Competition website provides the dataset that was provided for each phase. The Spearman rank correlation coefficient, which assesses how closely the rankings predicted by the system match human assessments, is the official evaluation statistic for this activity. The GitHub page8 dedicated to the competition has the assessment script for this common job, which offers a uniform process for rating the effectiveness of competing solutions. The Spearman correlation coefficient can be calculated using the formula found in
$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2-1)}$

where n is the number of samples and d is the pairwise distances between the ranks of the variables xi and yi.

Table 2: Performance of Models Used

Model	BERT-BASE
Supervised	0.803
UnSupervised	0.693

Results for our experiments are shown in the table-2. It is important to remember that the bert-base-nli-mean-tokens model is no longer in use because of its poorer sentence embeddings.But In Supervised Approach, after finetuning the model with training data , the sentence embeddings are more meaningful as shown in fig-1. We were able to achieve better results than baseline in Unsupervised Approach using this model. In Supervised Approach , the baseline score is 0.830 and our proposed approach score is 0.803, with further improvements to our approach, we might achieve better results than baseline.

6 Conclusion

In order to address Task 1 at SemEval-2024, this paper presents the use of a BERT-BASE model embedding. For our submission, we chose a Supervised (finetuning) and unsupervised (not finetuning) approach, utilizing pre-trained Transformers that are already tailored to the domain. Based on this strategy, we used the contextual embeddings generated by the Sentence Transformer and used cosine similarity to measure the similarity between pairs of sentences, thereby quantifying the similarity between them. Although our method was successful, there is still room for improvement, as evi-

denced by the final ranking. Possible alternate approaches include utilizing the zeroshot capabilities of models like GPT , increasing the training data size by adding more datasets. There is some space for improvement in our straightforward method when compared to the top-performing models. It is noteworthy, nonetheless, that the assignment could be completed with a reasonable computational cost and no further pre-training thanks to Google Colab's free online tools.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *arXiv preprint arXiv:2110.04845*.

Dhivya Chandrasekaran and Vijay Mago. 2021. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37.

Jacob Devlin et al. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

M A K Halliday and R Hasan. 1976. *Cohesion in English*. Longman.

Mena Hany et al. 2023. Enhancing textual relatedness assessment with combined transformers-embedding similarity techniques and machine learning regressors. In *Intelligent Methods, Systems, and Applications (IMSA)*. IEEE.

Tengfei Liu et al. 2023. Zero-shot text classification with semantically extended textual entailment. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *Preprint*, arXiv:2402.08638.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

TECHSSN at SemEval-2024 Task 1: Multilingual Analysis for Semantic Textual Relatedness using Boosted Transformer Models

Shreejith Babu G, Ravindran V, Aashika Jetti
Rajalakshmi Sivanaiah, Angel Deborah S
Department of Computer Science and Engineering
Sri Sivasubramaniya Nadar College of Engineering
Chennai - 603110, Tamil Nadu, India
{shreejithbabu2213006, ravindran2213003, aashika2210193}@ssn.edu.in,
{rajalakshmis, angeldeborahs}@ssn.edu.in

Abstract

This paper presents our approach to SemEval-2024 Task 1: Semantic Textual Relatedness (STR). Out of the 14 languages provided, we specifically focused on English and Telugu. Our proposal employs advanced natural language processing techniques and leverages the Sentence Transformers library for sentence embeddings. For English, a Gradient Boosting Regressor trained on DistilBERT embeddings achieves competitive results, while for Telugu, a multilingual model coupled with hyperparameter tuning yields enhanced performance. The paper discusses the significance of semantic relatedness in various languages, highlighting the challenges and nuances encountered. Our findings contribute to the understanding of semantic textual relatedness across diverse linguistic landscapes, providing valuable insights for future research in multilingual natural language processing.

1 Introduction

Semantic Textual Relatedness (STR) is a pivotal aspect of natural language processing (NLP) that underlies the foundation of various language-related tasks (Ousidhoum et al., 2024a,b). This task takes this challenge to a global scale by encompassing 14 languages, including Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Morrocan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu. This multilingual approach transcends linguistic boundaries, fostering collaboration and research within the NLP community. Exploring a diverse array of languages in this endeavor encourages the development of models adept at handling the distinct linguistic nuances inherent in each language. This progress fosters a more inclusive and universally applicable approach to natural language processing research.

SemEval-2024 Task 1 delves into the automated detection of semantic relatedness between pairs of sentences, a foundational aspect for unraveling meaning. The task's embrace of multiple languages is crucial, encompassing a spectrum of linguistic characteristics. This inclusivity fosters a collaborative atmosphere for researchers, pushing them to craft models adept at capturing semantic nuances across diverse linguistic landscapes. The task's importance extends to the assessment and benchmarking of sentence representation methods, pivotal for numerous NLP applications. STR evaluated through tasks, play a crucial role in areas such as question answering, summarization, and information retrieval. The task's outcomes serve as a benchmark, guiding the development and refinement of models that can effectively discern the relatedness of sentences, regardless of language.

When approaching this task, we concentrated on two languages: English and Telugu. Employing the Sentence Transformers library, we make use of the DistilBERT model for generating sentence embeddings in English and opt for a multilingual model for handling Telugu. The choice of state-of-the-art models and embedding techniques underscores our commitment to developing robust solutions capable of handling the linguistic diversity presented in this task.

Participating in this task has revealed crucial insights into the intricacies of semantic relatedness across languages. Our system demonstrated better performance, particularly in English, achieving noteworthy Spearman correlation coefficients on the test set. However, the challenges surfaced in capturing subtle nuances in semantic relations, especially in the context of Telugu. This highlights the necessity for specialized methodologies to address the complexities of multilingual semantic relatedness. As we explore the methodology, experiments, and results in the following sections, we delve deeper into the intricacies of our approach, providing a comprehensive understanding of how our model navigates the challenges posed by the

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 907–912

task.

2 Background

The task at hand revolves around Semantic Textual Relatedness (STR), focusing on evaluating the degree of semantic closeness between sentences in both English and Telugu. Unlike emotion recognition, this task delves into understanding the relationships between sentences rather than categorizing emotions in code-mixed interactions. In this scenario, the input comprises pairs of sentences in either English or Telugu, and the objective is to determine the relatedness score for each pair on a scale from 0 to 1.

For instance, consider the following English sentence pair:
Sentence 1: "The sun is setting over the horizon, casting a warm glow on the city."
Sentence 2: "As the day comes to an end, the sun sets, and the city is bathed in a warm glow."
Score Label: 0.85
In this example, the relatedness score of 0.85 indicates a high degree of semantic closeness between the two sentences, as they convey similar information about the sunset and the warm glow of the city.
Sentence 1: "The scientific method involves systematic observation and experimentation."
Sentence 2: "Bicycles are a popular mode of transportation in urban areas."
Score Label: 0.15

In this example, the relatedness score of 0.15 indicates a low degree of semantic closeness between the two sentences. Similarly, a dataset exists for Telugu.

The datasets used for this task include pairs of sentences in English and Telugu, capturing the real-world scenario of diverse linguistic interactions. These datasets are annotated with relatedness scores, providing a basis for training and evaluating models effectively. The input parameters consist of the sentence pairs, and the output involves predicting the relatedness score for each pair. The multilingual nature of the task fosters collaboration and research across linguistic boundaries, contributing to a more inclusive and globally applicable approach in the NLP community.

3 Related Work

Palakorn Achananuparp et. al.presents an evaluation of fourteen existing text similarity measures (Achananuparp et al., 2008). The ability to accurately judge the similarity between natural language sentences is crucial for various applications such as text mining, question answering, and text summarization. The evaluation encompasses three different datasets: TREC9 question variants, Microsoft Research paraphrase corpus, and the third recognizing textual entailment dataset. The study explores three classes of measures: word overlap, TF-IDF, and linguistic measures. The goal is to judge sentence pairs based on the notion that they have identical meanings, considering factors such as paraphrase or entailment. They address the challenges of computing sentence similarity, highlighting the importance of recognizing semantic equivalence beyond surface form comparisons.

Pantulkar Sravanthi and B. Srinivasu, address the challenge of measuring sentence similarity, emphasizing the importance of semantic similarity over syntactic measures (Sravanthi and Srinivasu, 2017). They introduce three semantic similarity approaches—cosine similarity, path-based (Wu–Palmer and shortest path), and feature-based. The feature-based approach incorporates WordNet, tagging, and lemmatization, showing superior performance in generating semantic scores. This study contributes valuable insights into semantic similarity measures and can enhance the understanding of feature-based approaches based on WordNet in sentence categorization.

Syed S. Akhtar et al. (Akhtar et al., 2017) address the need for word similarity datasets in Indian languages, specifically Urdu, Telugu, Marathi, Punjabi, Tamil, and Gujarati. They introduce manually annotated monolingual word similarity datasets for these languages, created through translation and re-annotation of English datasets. The paper presents baseline scores for word representation models using state-of-the-art techniques for Urdu, Telugu, and Marathi, evaluated on the newly created datasets. This work contributes valuable resources for evaluating word representations in Indian languages, fostering the development of techniques leveraging word similarity.

4 System Overview

To optimize efficiency, we methodically integrated numerous critical algorithms and modeling decisions into our semantic textual relatedness model.

4.1 Data Preprocessing

4.1.1 Text Cleaning

In the initial phase of our preprocessing pipeline, (Kadhim, 2018) we address the cleanliness of the textual data for both Telugu and English. For Telugu, we employ a language-specific approach, utilizing a tokenizing function tailored to the Telugu script. This ensures the proper segmentation of words while also excluding unwanted elements such as punctuation, special characters, and digits. Similarly, for English, we apply standard tokenization techniques to achieve a clean and well-structured representation, eliminating extraneous symbols and numerical values. This initial cleaning step lays the foundation for subsequent language-specific processing.

4.1.2 Language-specific Tokenization

Recognizing the distinct linguistic features of Telugu and English, we implement language-specific tokenization methods. In the case of Telugu, we adapt tokenization to the unique script and structural characteristics of the language. This approach ensures the accurate representation of Telugu text for downstream tasks. Conversely, for English, we rely on conventional tokenization techniques suited for the Latin script. By tailoring tokenization to the linguistic attributes of each language, we pave the way for more effective and contextually rich representations in subsequent stages of the preprocessing pipeline.

4.1.3 Stop Word Removal

Stopwords were removed from the combined tokens in order to enhance the model's focus on pertinent content. Through the removal of noise and refinement of the raw data, a deeper comprehension of the underlying sentiment was made possible.

4.1.4 Data Splitting

The `train_test_split` function from the `scikit-learn` library was used to split the preprocessed data into training and testing sets. This made it possible to thoroughly assess the model's capacity for generalization using data that had never been seen before.

4.2 Model Architecture

4.2.1 English

Embedding with DistilBERT: To capture semantic meanings, we utilized the DistilBERT model (version: distilbert-base-uncased) (Kici et al., 2021) from the Sentence Transformers library to generate sentence embeddings. DistilBERT is a distilled version of the BERT model, designed for faster inference while maintaining competitive performance. Sentences were encoded into embeddings using DistilBERT, facilitating the creation of robust representations. These embeddings served as the input features for subsequent relatedness score prediction.

Gradient Boosting Regressor Model:

To model the relatedness scores, we employed the Gradient Boosting Regressor algorithm. Specifically, we utilized the GradientBoostingRegressor class from the scikit-learn library (version: 0.24.2). The model was trained on the encoded sentences and evaluated on the test set.

In the process of hyperparameter tuning, the following parameters were optimized:
Learning rate: 0.05
Number of estimators: 200
Maximum depth of each estimator (max depth): 3
Subsample ratio of the training instances (subsample): 0.8

These hyperparameters were chosen based on a grid search conducted to maximize the model's effectiveness in predicting semantic textual relatedness for English sentences. Specifically, the learning rate controls the contribution of each tree in the ensemble, while the number of estimators determines the number of boosting stages. Additionally, the maximum depth of each tree and the subsample ratio influence the depth of the individual trees and the sampling strategy, respectively.

Model Persistence and Reporting: The trained Gradient Boosting Regressor model is saved for future use. Spearman correlation on the test set provides a quantitative measure of the model's ability to predict sentence relatedness. The model's performance, including the correlation coefficient, is printed for further analysis.

4.2.2 Telugu

Multilingual Sentence Embeddings: For Telugu, we opt for a pre-trained multilingual model (paraphrase-multilingual-MiniLM-L12-v2) to generate sentence embeddings. The Telugu dataset is encoded into embeddings using this multilingual model. The Telugu model is trained using a Gradient Boosting Regressor, and its performance is

evaluated on the test set using the Spearman correlation coefficient.

Advanced Model Tuning: We used an approach similar to one used for the English dataset where hyperparameter tuning is performed using a grid search for the Gradient Boosting Regressor on Telugu data. The best model is selected based on the optimal combination of hyperparameters, leading to improved performance. The chosen model is subsequently applied for prediction and evaluation.

4.3 Model Evaluation

4.3.1 Performance Metrics

To gauge the performance of the English-relatedness detection model, we utilized a comprehensive set of metrics, incorporating the Spearman correlation coefficient. These metrics collectively offered a well-rounded understanding of the model's accuracy, precision, and capacity to capture the subtleties of relatedness among English sentences. Following a similar evaluation approach for the Telugu-relatedness detection model, we subjected it to a thorough assessment using performance metrics. The Spearman correlation coefficient served as a valuable measure to assess the accuracy and precision of the model in capturing relatedness between Telugu sentences.

4.3.2 Prediction on the Test Set

We employed the trained model to predict textual relatedness on a test set. This involved passing the preprocessed test data through the model and deciphering the anticipated labels for subsequent analysis.

5 Experimental Setup

5.1 Data Preprocessing for Subtask 1a (English)

For the English dataset, we adopted a two-step preprocessing approach. Initially, sentences underwent precise tokenization using the Sentence Transformers library. This process harnessed the advanced capabilities of DistilBERT to encode sentences into dense embeddings, establishing the foundation for subsequent model training. The selected Gradient Boosting Regressor model underwent training on these embeddings, contributing to improved semantic textual relatedness.

5.2 Data Preprocessing for Subtask 1b (Telugu)

For Telugu, we implemented dedicated preprocessing, involving nuanced tokenization facilitated by the Natural Language Toolkit (NLTK). This process was complemented by a careful removal of stopwords, specifically tailored for Telugu text. Subsequently, the (Gillioz et al., 2020) Sentence Transformer's multilingual model came into play, encoding Telugu sentences into high-dimensional embeddings. These embeddings formed the basis for training a Gradient Boosting Regressor model. Importantly, hyperparameter tuning played a pivotal role in fine-tuning the model's performance specifically for Telugu.

5.3 Hyperparameter Tuning

For Subtask 1a, our focus on hyperparameter tuning aimed to optimize the performance of the Gradient Boosting Regressor model on the English dataset. The primary goal was to fine-tune the model for improved semantic textual relatedness prediction.

In Subtask 1b, specifically for Telugu, we conducted a thorough hyperparameter tuning process using GridSearchCV. Key hyperparameters such as estimators, learning rate, max-depth, and subsample were carefully explored to enhance the model's efficacy. This step was intended to fine-tune the Gradient Boosting Regressor model for optimal performance on the Telugu dataset.

5.4 External Tools / Libraries

External tools and libraries played a pivotal role in our experimentation. Sentence Transformers (v2.0.0) with its sophisticated capabilities was instrumental in encoding sentences into high-dimensional embeddings. NLTK (v3.6.3) facilitated precise tokenization and stopword removal, contributing to meticulous linguistic preprocessing. Scikit-Learn (v0.24.2) emerged as the preferred library for machine learning models and hyperparameter tuning, providing a standardized and comprehensive experimental framework.

5.5 Evaluation Metric

The evaluation metric of choice was the Spearman correlation coefficient. Renowned for its ability to discern the monotonic relationship between predicted scores and gold standard scores, this metric offered a nuanced assessment of semantic textual relatedness.

Approach	Accuracy
distilbert-base-uncased (English)	0.57
paraphrase-multilingual-MiniLM-L12-v2 (Telugu)	0.527

Table 1: Comparison of Accuracy for Different Approaches

6 Results

The system's performance was assessed using a regression model that predicts similarity scores between English sentences. The Spearman Correlation Coefficient is the major quantitative finding, measuring the monotonic relationship between expected and actual similarity scores. The Spearman correlation coefficient, often denoted as (rho), is a statistical measure used to assess the strength and direction of the monotonic relationship between two variables. Specifically, in the context of evaluating models, the Spearman coefficient is employed to quantify the association between predicted scores and actual scores.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (1)$$

Table 1 shows the Spearman score of the models used for English and Telugu. The English model, employing advanced natural language processing techniques and utilizing DistilBERT embeddings, demonstrated noteworthy performance on the test set. The Spearman correlation coefficient, a key indicator of the model's ability to predict relatedness between English sentences, was calculated. Our model achieved a Spearman correlation coefficient of 0.57 on the test set, showcasing its effectiveness in capturing semantic nuances and relationships within English sentence pairs.

The model's performance is influenced by the quality of the preprocessing applied to the text data. Any limitations or challenges encountered during the preprocessing stage, such as handling rare words or specific language nuances, should be discussed. The performance of our Telugu model in predicting semantic relatedness scores using a fine-tuned Gradient Boosting Regressor with hyperparameter tuning and embeddings from the Sentence Transformer model 'paraphrase-multilingual-MiniLM-L12-v2' is evaluated here. On the test set, our Telugu model achieved a Spearman correlation coefficient of 0.527. This result signifies a strong positive monotonic relationship between the predicted relatedness scores and the actual scores. The Spearman correlation coefficient is a crucial indicator of the model's ability to capture the underlying trends in sentence similarity within the Telugu language. The hyperparameter tuning process identified the following optimal hyperparameters for the Gradient Boosting Regressor on the Telugu dataset: learning rate : 0.05, max depth: 3, n_estimators: 200, subsample: 0.8 . These hyperparameters represent the configuration that maximizes the model's effectiveness in predicting relatedness scores for Telugu sentences.

7 Conclusion

Our system, anchored in the robust capabilities of Sentence Transformers and Gradient Boosting Regressor models, showcased a better performance in predicting semantic textual relatedness. The fusion of advanced tokenization, embeddings, and hyperparameter tuning resulted in a model finely attuned to the intricacies of the English and Telugu languages.

The results on the evaluation metrics, particularly the Spearman correlation coefficient, underscore the efficacy of our approach in capturing the nuances of semantic relatedness across diverse sentence pairs. The successful adaptation to multiple languages, as evident in the Telugu experiments, showcases the versatility of our system.

For future work, delving deeper into language-specific processing techniques and exploring more sophisticated models may unlock additional performance gains. Additionally, expanding the system's applicability to handle a broader array of languages and domains could further enhance its utility in real-world applications. Overall, our endeavors open avenues for continuous refinement and exploration in the realm of semantic textual relatedness prediction.

References

Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen. 2008. The evaluation of sentence similarity measures. In *Data Warehousing and Knowledge Discovery: 10th International Conference, DaWaK 2008 Turin, Italy, September 2-5, 2008 Proceedings 10*, pages 305–316. Springer.

Syed Sarfaraz Akhtar, Arihant Gupta, Avijit Vajpayee, Arjit Srivastava, and Manish Shrivastava. 2017.

Word similarity datasets for indian languages: Annotation and baseline systems. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 91–94.

Anthony Gillioz, Jacky Casas, Elena Mugellini, and Omar Abou Khaled. 2020. Overview of the transformer-based models for nlp tasks. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 179–183. IEEE.

Ammar Ismael Kadhim. 2018. An evaluation of preprocessing techniques for text classification. *International Journal of Computer Science and Information Security (IJCSIS)*, 16(6):22–32.

Derya Kici, Garima Malik, Mucahit Cevik, Devang Parikh, and Ayse Basar. 2021. A bert-based transfer learning approach to text classification on software requirements specifications. In *Canadian Conference on AI*, volume 1, page 042077.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *Preprint*, arXiv:2402.08638.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Pantulkar Sravanthi and B Srinivasu. 2017. Semantic similarity between sentences. *International Research Journal of Engineering and Technology (IRJET)*, 4(1):156–161.

Noot Noot at SemEval-2024 Task 7: Numerical Reasoning and Headline Generation

Sankalp Bahad[1]
IIIT Hyderabad
sankalp.bahad@research.iiit.ac.in

Yash Bhaskar[1]
IIIT Hyderabad
yash.bhaskar@research.iiit.ac.in

Parameswari Krishnamurthy[2]
IIIT Hyderabad
param.krishna@iiit.ac.in

Abstract

Natural language processing (NLP) models have achieved remarkable progress in recent years, particularly in tasks related to semantic analysis. However, many existing benchmarks primarily focus on lexical and syntactic understanding, often overlooking the importance of numerical reasoning abilities. In this paper, we argue for the necessity of incorporating numeral-awareness into NLP evaluations and propose two distinct tasks to assess this capability: Numerical Reasoning and Headline Generation. We present datasets curated for each task and evaluate various approaches using both automatic and human evaluation metrics. Our results demonstrate the diverse strategies employed by participating teams and highlight the promising performance of emerging models like Mixtral 8x7b instruct. We discuss the implications of our findings and suggest avenues for future research in advancing numeral-aware language understanding and generation.

1 Introduction

Natural language processing (NLP) models have achieved impressive performance on a wide range of semantic analysis tasks in recent years. However, the majority of benchmarks used to evaluate these models, including past SemEval shared tasks, have focused predominantly on lexical and syntactic understanding, with little emphasis on numerical reasoning abilities. In this paper, we argue that comprehending and reasoning with numerical values expressed in text is vital for robust language understanding, and should be an integral part of NLP evaluations going forward (num).

We demonstrate across several application scenarios that a lack of numeracy can undermine model performance and result in erroneous output. As an illustration, fine-grained sentiment analysis, as explored in SemEval-2017 Task 5, relies heavily on distinguishing subtle differences in sentiment intensity. Anticipating a 30% stock price increase implies a markedly more positive outlook than a 3% rise. Without accounting for the differential impact of these numbers, sentiment analysis models may fail to capture such nuances.

Similarly, in legal judgment prediction settings like SemEval-2023 Task 6, sentencing decisions can hinge on numerical quantities - stealing $100,000 typically incurs harsher penalties than stealing $10. Clinical inference use cases such as SemEval-2023 Task 7 also require sensitivity to numbers like blood pressure readings, where contrasts between 121 mmHg and 119 mmHg could indicate notably different health outlooks (Devlin et al., 2019).

These examples highlight the limitations of current benchmarking paradigms in evaluating true language comprehension. Accordingly, we propose that new numerically-grounded NLP tasks be developed to test numerical reasoning capacities. Recent work has begun exploring this direction, but substantial efforts are still needed to build robust models that demonstrate human-like numeracy. We outline a potential experimental framework and novel dataset for this purpose in the following sections.

2 Dataset

Task 3 (Huang et al., 2023) of our study comprises two distinct subtasks: numerical reasoning and headline generation. We describe the dataset for each subtask separately below:

2.1 Subtask 1: Numerical Reasoning

For the numerical reasoning subtask, we curated a dataset consisting of news headlines with missing numerical values. Each instance in the dataset includes a news article along with a headline where a numerical value is replaced with a blank. An example of the format for each instance is as follows:

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 913–917

- **News Article**: [Insert news article text here]
- **Headline with Blank**: "Study predicts a [blank] increase in global temperatures by 2050."
- **Target Value**: The correct numerical value that should fill in the blank.

The dataset includes a diverse range of news articles covering various topics such as climate change, economics, healthcare, and more. Each instance is associated with a target value representing the correct numerical answer.

2.2 Subtask 2: Headline Generation

For the headline generation subtask, we compiled a dataset consisting of news articles without headlines. Each instance in this dataset includes a news article, and the task is to generate a headline based on the content of the article. An example instance is provided below:

- **News Article**: [Insert news article text here]
- **Target Headline**: The headline that should be generated based on the content of the news article.

Similar to the numerical reasoning dataset, the articles cover a wide range of topics to ensure diversity in the generated headlines.

The table below shows the number of data points in the validation, test, and train sets for the Numerical Reasoning task:

Dataset	Validation	Test	Train
Numerical Reasoning	2572	4921	21157

Table 1: Dataset Statistics for Numerical Reasoning

Similarly, the table below presents the number of data points in the validation, test, and train sets for the Headline Generation task:

Dataset	Validation	Test	Train
Headline Generation	2365	5227	21157

Table 2: Dataset Statistics for Headline Generation

3 Methods

Zero-shot prompting is an effective method for these news headline tasks because it allows the model to apply its generalized language understanding capabilities to novel tasks without extensive fine-tuning. The model can deduce the appropriate responses based solely on the instructions and examples provided in the prompt.

The prompts are carefully engineered to provide the model with clear guidelines and context. For the numerical reasoning task, the prompt poses the incomplete headline as a question and asks the model to fill in the blank with only a numerical value. This focuses the model on extracting and inferring the relevant number from the article text.

Similarly, the headline generation prompt provides the news article as context and directly instructs the model to generate a headline summarizing the key information. The simplicity of these prompts allows the model to use its innate language skills to produce fitting responses without needing gradient-based training on the specific tasks.

Furthermore, the varied topics and contexts in the dataset require the model to adapt its numerical and summarization strategies across different situations. This tests the model's ability to generalize based on the prompt instructions, rather than overfitting to biases in a narrow dataset. The broad applicability demonstrated through zero-shot prompting highlights the versatile reasoning capacity gained through the model's pretraining.

Overall, zero-shot prompting is an elegant and effective approach for this study, as it allows assessment of the model's intrinsic skills at numerical deduction and text summarization when provided suitable prompts. The prompt formulation is key to eliciting successful performance without task-specific fine-tuning.

3.1 Subtask 1: Numerical Reasoning

The prompt (Zamfirescu-Pereira et al., 2023) used for determining the value of the missing numerical variable is as follows:

```
message = "News: {News}.\n
Headline: {Headline}\n\n
What is the value of ___?
Only give a numerical Response: "

prompt = f"[INST] {message} [/INST]"
```

3.2 Subtask 2: Headline Generation

For the headline generation subtask, participants are required to generate a headline based on the provided news article. The prompt for headline generation is as follows:

Example 1: Numerical Reasoning

News: (Oct 1, 2009 3:30 PM CDT) Want to catch up on YouTube2019s greatest hits but don2019t have the time? No problem: Just watch a new 4-minute mash-up that brings 100 of the best (or worst, depending on your viewpoint) together. Clips include such classics as Keyboard Cat and David After Dentist, and stars range from Obama Girl to the Dr. Pepper Guys, Time reports. Watch it at left.
Masked Headline: Watch 100 YouTube Classics in ____ Minutes
Calculation: Copy(4)
Ans: 4

Example 2: Numerical Reasoning

News: (Nov 16, 2009 8:40 AM) A rocket attack intended for a French general instead killed three children and wounded 20 others in a busy market northeast of Kabul today. Insurgents fired into the marketplace hoping to hit a meeting between Brig. Gen. Marcel Druart and tribal elders from Tagab Valley, where France is in the midst of a major offensive. Neither Druart nor any of his troops were harmed.
Masked Headline: Afghan Rocket Misses French General, Kills ____ Kids
Calculation: Trans(three)
Ans: 3

Example 3: Numerical Reasoning

News: (Mar 12, 2009 3:19 PM CDT) Stocks rose steadily after a morning dip today, with the Dow closing back over 7,000 points, MarketWatch reports. Bank of America and General Motors shot up 18.2% and 14.5%, respectively, after each announced they don2019t expect to ask the government for more bailout cash. The Dow ended up 239.66 at 7,170.06. The Nasdaq rose 54.46, settling at 1,426.10; the S
P 500 closed up 29.38 at 750.74.
Masked Headline: Dow Up 240, Retakes ____K Mark
Calculation: Paraphrase(7,000,K)
Ans: 7

Example 4: Headline Generation

News: (Mar 25, 2009 12:30 PM CDT) What's Italian for leadfoot ?00a0A Milanese man going 168mph was busted on four separate highway cameras in less than hour, ANSA reports. He was driving for his employer, whose lawyers argue that he should be responsible for just one infraction. They said they also plan to cite a court ruling that says signs identifying cameras must be a certain distance from a speed trap, adding: Naturally, we do not condone such driving at all.
Headline: Italian Going 168mph Gets 4 Tickets in 1 Hour

Example 5: Headline Generation

News: (Apr 21, 2010 12:51 PM CDT) The $100 bill is getting a new look and two high-tech security features to curb counterfeiters, the AP reports. A 3D security ribbon on the front has images of bells and 100s that move as you tilt the bill. The note, which is out next February, also has a Liberty Bell that seems to disappear. The government has more details here, along with a video that borders on the cheesy side here.
Headline: $100 Bill Goes 3D

```
message = "News: {News}.
Generate a headline based on the provided
news article: "

prompt = f"[INST] {message} [/INST]"
```

4 Results

4.1 Numerical Reasoning

The table 3 presents the results for the Numerical Reasoning task:

Rank	Team	Score
1	CTYUN-AI	0.95
2	zhen qian	0.94
3	YNU-HPCC	0.94
4	NCL_NLP	0.94
5	NumDecoders	0.91
6	Infrrd.ai	0.90
7	hc	0.88
8	NLPFin	0.86
9	NP-Problem	0.86
10	AlRah	0.83
11	Noot Noot	0.77
12	GPT-3.5	0.74
13	Sina Alinejad	0.74
14	StFX-NLP	0.60

Table 3: Numerical Reasoning Results

In the domain of Numerical Reasoning, we showed a commendable performance, achieving a score of 0.77, marginally surpassing the baseline score attributed to the GPT-3.5 model, which stood at 0.74.

4.2 Headline Generation

4.2.1 Auto Evaluation

The table 4 presents the results for the Headline Generation task based on auto evaluation metrics.

We demonstrated notable proficiency in headline generation, attaining a ROUGE score of 38.4, a BERT score of 57.5, and a Mover's Accuracy score of 3.6 in the automated evaluation.

4.2.2 Human Evaluation

The table 5 presents the results for the Headline Generation task based on human evaluation metrics.

Human evaluators accorded the team a score of 1.68, indicating favorable reception of the generated headlines.

Team	Num Acc.	ROUGE	BERT Score	Mover Score
ClusterCore	38.2	51.6	13.9	33.5
Noot Noot	38.4	57.5	3.6	31.5
Infrrd.ai	65.8	68.4	61.3	46.8
np_problem	73.5	76.9	67.3	39.8
hinoki	62.4	66.3	55.2	43.1
Challenges	73.0	82.2	56.2	31.2
NCL_NLP	62.1	65.5	55.9	43.5
YNU-HPCC	69.0	73.0	61.8	48.9
NoNameTeam	55.7	57.7	52.1	40.7

Table 4: Auto Evaluation Results for Headline Generation

Team	Num Acc. (50 Headlines)	Recommendation(100 News)
ClusterCore	1.60	31
Noot Noot	1.68	11
Infrrd.ai	1.81	22
np_problem	1.57	14
hinoki	1.67	16
Challenges	1.70	10
NCL_NLP	1.73	16
YNU-HPCC	1.69	15
NoNameTeam	1.59	12

Table 5: Human Evaluation Results for Headline Generation

5 Conclusion

In this study, we explored the performance of various approaches for two distinct tasks: Numerical Reasoning and Headline Generation. Across both tasks, we observed a range of performances among the participating teams, indicating the diversity of techniques and strategies employed.

For Numerical Reasoning, the top-performing teams demonstrated high accuracy and effective reasoning capabilities, leveraging a combination of techniques to achieve superior results. Notably, the utilization of advanced models and fine-tuning methodologies played a crucial role in enhancing performance on this task.

In Headline Generation, both auto and human evaluations highlighted the effectiveness of certain teams in generating accurate and engaging headlines. Teams employing sophisticated natural language processing techniques, such as advanced neural models and feature engineering, exhibited superior performance in generating headlines that

resonated well with both automated evaluation metrics and human judges.

Furthermore, the introduction of Mixtral 8x7b instruct model showcased promising capabilities in both tasks. Despite not being fine-tuned specifically for the tasks at hand, the Mixtral model demonstrated competitive performance, particularly in Headline Generation. This suggests the robustness and versatility of the Mixtral 8x7b instruct model in understanding and generating natural language content across diverse domains and tasks.

Overall, our findings underscore the importance of leveraging state-of-the-art models and techniques in tackling complex natural language processing tasks. Additionally, the emergence of pretrained models like Mixtral 8x7b instruct offers promising avenues for future research and development, as they provide strong baselines and require minimal fine-tuning to achieve competitive performance across various NLP tasks.

References

Semeval-2024 task 7: Numeral-aware language understanding and generation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21.

NootNoot at SemEval-2024 Task 8: Fine-tuning Language Models for AI vs Human Generated Text detection

Sankalp Bahad[1]
IIIT Hyderabad
sankalp.bahad@research.iiit.ac.in

Yash Bhaskar[1]
IIIT Hyderabad
yash.bhaskar@research.iiit.ac.in

Parameswari Krishnamurthy[2]
IIIT Hyderabad
param.krishna@iiit.ac.in

Abstract

In this paper, we introduce a machine-generated text detection system designed to tackle the challenges posed by the proliferation of large language models (LLMs). With the rise of LLMs such as ChatGPT and GPT-4, there is a growing concern regarding the potential misuse of machine-generated content, including misinformation dissemination. Our system addresses this issue by automating the identification of machine-generated text across multiple subtasks: binary human-written vs. machine-generated text classification, multi-way machine-generated text classification, and human-machine mixed text detection. We employ the RoBERTa Base model and fine-tune it on a diverse dataset encompassing various domains, languages, and sources. Through rigorous evaluation, we demonstrate the effectiveness of our system in accurately detecting machine-generated text, contributing to efforts aimed at mitigating its potential misuse.

1 Introduction

Large language models (LLMs) are becoming mainstream and easily accessible, bringing in an explosion of machine-generated content over various channels, such as news, social media, question-answering forums, educational, and even academic contexts. Recent LLMs, such as ChatGPT and GPT-4, generate remarkably fluent responses to a wide variety of user queries. The articulate nature of such generated texts makes LLMs attractive for replacing human labor in many scenarios. However, this has also resulted in concerns regarding their potential misuse, such as spreading misinformation and causing disruptions in the education system. Since humans perform only slightly better than chance when classifying machine-generated vs. human-written text, there is a need to develop automatic systems to identify machine-generated text with the goal of mitigating its potential misuse.

The advent of sophisticated large language models (LLMs), including ChatGPT and GPT-4, has catalyzed a surge in artificially generated text across myriad domains, from news media to social platforms, educational resources, and scholarly publications. These neural network models exhibit an unprecedented capacity to produce natural language, enabling the automation of written content creation. However, the human-like fluency of LLM outputs has concurrently raised serious concerns surrounding potential misuse.

With syntactically coherent and topically relevant text, LLMs could plausibly disseminate misinformation, plagiarize or falsify documents, and automate persuasion-based attacks on a massive scale. The integration of models like ChatGPT into education has additionally ignited fierce debate; while proponents highlight opportunities for personalized instruction, critics argue LLMs enable academic dishonesty and undermine human knowledge acquisition. Amidst this controversy, institutions urgently seek policies to uphold academic integrity.

Alarmingly, humans perform only marginally better than random chance at distinguishing machine-generated versus human-authored text. Developing reliable technical systems to automatically detect AI content has therefore become a research priority. The goal is to provide educators, moderators, and end users tools to identify LLM outputs, thereby mitigating potential dangers from increasingly accessible, human-like models.

Constructing robust LLM detectors demands interdisciplinary collaboration, combining machine learning advances with insights from fields like ethics, media studies, and education. With judicious coordination across stakeholders, experts aim to actualize benefits of LLMs for automation while curtailing risks of misinformation, deception, and cheating.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 918–921

2 Dataset

The dataset provided by the organizers of this shared task (Wang et al., 2024a) comprises a diverse collection of texts encompassing various domains, languages, and sources. The dataset is structured to address the three subtasks outlined: binary human-written vs. machine-generated text classification (Subtask A), multi-way machine-generated text classification (Subtask B), and human-machine mixed text detection (Subtask C).

For Subtask A (Binary Classification), the dataset consists of a balanced corpus of human-written and machine-generated texts. The human-written texts are sourced from various publications, academic papers, forums, and social media platforms. The machine-generated texts are generated by state-of-the-art language models such as ChatGPT, GPT-4, cohere, davinci, bloomz, and Dolly. These texts cover a wide range of topics to ensure diversity and representativeness.

For Subtask B (Multi-Way Classification), the dataset includes texts generated by each of the six specified language models: ChatGPT, cohere, davinci, bloomz, Dolly, and human-written texts. The texts are annotated to indicate their respective sources, enabling the classification task to determine the origin of each text accurately.

For Subtask C (Human-Machine Mixed Text Detection), the dataset contains texts where the first part is human-written, and the subsequent part is machine-generated. Annotations demarcate the boundary where the transition from human to machine-generated text occurs. This allows for training and evaluating models on detecting the boundary between human and machine-generated segments within a single text.

The dataset is preprocessed to remove noise, standardize formatting, and ensure consistency across texts.

Subtask	Train	Dev
A (Monolingual)	119,757	5,000
A (Multilingual)	172,417	4,000
B	71,027	3,000
C	3,649	505

Table 1: Dataset Statistics for Each Subtask

3 Methods

We employed a fine-tuning approach using the XLM-RoBERTa Base model (Conneau et al., 2020) for the task of machine-generated text detection.

We chose Roberta-base as our base model for fine tuning as XLM-RoBERTa is a multilingual language model optimized for classification tasks. It is pretrained on massive multilingual data, and has a robust architecture and performance enable efficient fine-tuning across diverse text classification problems with state-of-the-art accuracy

The fine-tuning process involves initializing the RoBERTa Base model with pre-trained weights and then fine-tuning it on our specific dataset for the tasks of binary human-written vs. machine-generated text classification (Subtask A), multi-way machine-generated text classification (Subtask B), and human-machine mixed text detection (Subtask C).

During fine-tuning, we optimize the model's parameters using stochastic gradient descent (SGD) with backpropagation. We employ task-specific loss functions, such as cross-entropy loss for classification tasks and mean squared error (MSE) for mixed text detection. Additionally, we utilize techniques such as dropout regularization to prevent overfitting and gradient clipping to stabilize training.

The RoBERTa Base model is fine-tuned separately for each subtask, with hyperparameters tuned using grid search or random search techniques. We split the dataset into training, validation, and test sets to facilitate model training and evaluation, ensuring that the model generalizes well to unseen data.

4 Results

We present the performance metrics achieved by our machine-generated text detection system on each of the subtasks: binary human-written vs. machine-generated text classification (Subtask A), multi-way machine-generated text classification (Subtask B), and human-machine mixed text detection (Subtask C). The evaluation metrics include F1 score (macro and micro) and accuracy.

4.1 Subtask A: Binary Classification

Epoch	F1 Macro	F1 Micro	Accuracy
1	0.85431	0.85463	0.85463
2	0.81726	0.81918	0.81918
3	0.80595	0.80859	0.80859

Table 2: Performance Metrics for Subtask A (Monolingual)

Epoch	F1 Macro	F1 Micro	Accuracy
1	0.65693	0.69128	0.69128
2	0.71308	0.72564	0.72564
3	0.64664	0.68958	0.68958

Table 3: Performance Metrics for Subtask A (Multilingual)

From the tables of results of Subtask A, we can observe that in Monolingual case we get a better accuracy and F1-Score. The scores are in the range of 0.8 to 0.85, which decrease with the increase in number of epochs. Hence, the best score is observed in the model trained only for 1 epoch. This pattern indicates that the model can possibly be overfitting on the data.

In case of Multilingual, we observe the best scores in the model trained for 2 epochs.

4.2 Subtask B: Multi-Way Classification

Epoch	F1 Macro	F1 Micro	Accuracy
1	0.80686	0.8065	0.8065
2	0.85083	0.851	0.851
3	0.83146	0.83117	0.83117
4	0.84295	0.84328	0.84328
5	0.86936	0.86794	0.86794

Table 4: Performance Metrics for Subtask B

From the results of Subtask B, we can observe that the model trained for epoch 5 performs the best. Based on the Micro and Macro F1 scores in the table, we can observe that since the Macro F1 increasing over epochs indicates the model is improving at predicting each individual class correctly. The Micro F1 is also increasing which suggests that overall predictive capability on the aggregate data is improving. However, Micro F1 can be influenced by performance on majority classes.

4.3 Subtask C: Human-Machine Mixed Text Detection

Epoch	MSE
1	63.13998
2	33.09197
3	28.01411
4	30.04774
5	27.12254

Table 5: Performance Metrics for Subtask C

From the results of Subtask C, we can observe that the provided mean squared error (MSE) values indicate the model loss decreased with each epoch of training from an initial value of 63.13998 at epoch 1 to 27.12254 at epoch 5. The difference in MSE between epoch 1 and 2 implies that the model is overfitting on the training data. The lowest MSE score is observed in Epoch 5.

5 Conclusion

In this study, we proposed a machine-generated text detection system capable of addressing three subtasks: binary human-written vs. machine-generated text classification (Subtask A), multi-way machine-generated text classification (Subtask B), and human-machine mixed text detection (Subtask C).

Our system leverages the RoBERTa Base model, fine-tuned on a diverse dataset comprising texts from various domains, languages, and sources. Through extensive experimentation and evaluation, we achieved promising results (Wang et al., 2024b) across all subtasks.

For Subtask A, our system demonstrated robust performance in distinguishing between human-written and machine-generated texts, achieving high F1 scores and accuracy across multiple epochs. Similarly, in Subtask B, where the classification involves identifying the source language model among multiple candidates, our system achieved competitive performance, indicating its effectiveness in multi-way classification scenarios.

In Subtask C, where the objective is to detect boundaries between human-written and machine-generated segments within a single text, our system showed reasonable performance, albeit with some room for improvement. Future work could focus on refining the model architecture and exploring additional features to enhance the system's performance in this challenging task.

Overall, our study highlights the importance and feasibility of developing automatic systems for detecting machine-generated text, contributing to efforts aimed at mitigating the potential misuse of large language models in various contexts.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised

cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

eagerlearners at SemEval2024 Task 5: The Legal Argument Reasoning Task in Civil Procedure

Hoorieh Sabzevari, Mohammadmostafa Rostamkhani, Sauleh Eetemadi
Iran University of Science and Technology
h_sabzevari@elec.iust.ac.ir, mo_rostamkhani97@comp.iust.ac.ir, sauleh@iust.ac.ir

Abstract

This study investigates the performance of the zero-shot method in classifying data using three large language models, alongside two models with large input token sizes and the two pre-trained models on legal data. Our main dataset comes from the domain of U.S. civil procedure. It includes summaries of legal cases, specific questions, potential answers, and detailed explanations for why each solution is relevant, all sourced from a book aimed at law students. By comparing different methods, we aimed to understand how effectively they handle the complexities found in legal datasets. Our findings show how well the zero-shot method of large language models can understand complicated data. We achieved our highest F_1 score of 64% in these experiments.

1 Introduction

Becoming skilled at presenting a legal case is essential for aspiring lawyers. It requires understanding not only the relevant legal areas but also using advanced reasoning tactics like making analogies and spotting hidden contradictions. (Chalkidis et al., 2022). Despite efforts to set standards for modern NLP models in legal language understanding, there still aren't complex tasks focusing on argumentation in legal matters.(Bongard et al., 2022)

The dataset utilized in this study was gathered from *The Glannon Guide To Civil Procedure* by Joseph Glannon (Glannon, 2018) in English. Each sample within the dataset comprises a question, a solution, and an introduction elaborating on the provided solution. The objective is to identify whether the given answer, derived from the introduction text, accurately addresses the question.

This paper explores various approaches to address the challenge of handling lengthy and intricate data, which can be challenging for human comprehension. Initially, we evaluated two models—Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2021) —known for their effectiveness in classifying data with large input tokens. Subsequently, we assessed the performance of two pre-trained models, Legal-RoBERTa (Chalkidis* et al., 2023) and Legal-XLM-RoBERTa (Niklaus et al., 2023) for legal data using the original code. Finally, we compared the performance of three large language models —GPT 3.5, Gemini, and Copilot— using the zero-shot method on the test dataset.

We recognized the significant impact of leveraging the capabilities and extensive capacity of large language models on analyzing data, especially those focusing on specific topics or lengthy content. Looking ahead, our goal is to improve prompts further to achieve superior results not only for this task but also for similar works. Further details regarding the implementation can be found in this GitHub repository.

2 Background

2.1 Task Setup

As previously mentioned, the original dataset for this task is sourced from *The Glannon Guide To Civil Procedure*. Each sample within the dataset comprises the following components:

1. Question 2. Answer 3. Label 4. Analysis 5. Complete Analysis 6. Explanation

It's noteworthy that "Analysis" and "Complete Analysis" were absent in the test data. The data split involves allocating the initial 80% of questions from each chapter to the training set, the subsequent 10% to the validation set, and the final 10%—typically more challenging questions—to the test set. The final dataset consists of 848 entries.

In this task, the inputs to the model consist of the "Question," "Explanation," and "Answer" values, while the output of the model is represented by the "Label." If the model determines that the

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 922–926

answer provided for the question aligns with the explanation, the output label will be 1; otherwise, it will be 0. The objective of this task is to assess the model's reasoning capabilities, particularly its ability to analyze legal issues effectively.

2.2 Related Work

2.2.1 Pre-trained Legal Language Models

Numerous studies have been conducted in the realm of legal issues. Given the challenges posed by comprehending lengthy texts within this domain using existing models, pre-trained language models have been tailored to address this need. One such model is the Legal-BERT model (Chalkidis et al., 2020), which is also employed in this paper. This study introduces a specialized model aimed at facilitating NLP-based legal research by fine-tuning the original BERT model for legal applications. (Li et al., 2023) introduces SAILER, a novel pre-trained linguistic model with a unique architecture designed for legal case retrieval. Furthermore, (Cui et al., 2023b) provides a comprehensive survey of existing Legal Judgment Prediction (LJP) tasks, datasets, and models within the legal domain, encompassing an overview of 8 pre-trained models across 4 languages as part of the LJP. Moreover, an end-to-end methodology is introduced by (Louis et al., 2023) for generating long-form answers to statutory law questions, addressing limitations in existing Legal Question Answering (LQA) approaches.

2.2.2 Domain-Specific LLMs in Law

A Large Language Model (LLM), such as ChatGPT, is remarkable for its ability to handle general-purpose language generation and a variety of other NLP tasks. Domain-specific LLMs are versatile models optimized to excel at specific tasks defined by organizational standards. They further empower lawyers to expand their understanding and explore specialized legal domains. For instance, (Colombo et al., 2024) is the first LLM designed explicitly for legal text comprehension and generation with 7 billion parameters. To empower the legal field, (Cui et al., 2023a) presents ChatLaw, an open-source legal LLM built with a high-quality, domain-specific fine-tuning dataset. The focus of (Savelka et al., 2023) is evaluating GPT-4's effectiveness in generating explanations for legal terms – specifically, whether they are accurate, clear, and relevant to the surrounding legislation.

3 System Overview

3.1 Preprocessing Data

Our dataset comprises 666 samples for the training set, 84 samples for the validation set, and 98 samples for the test set. Initially, we excluded the columns "Analysis" and "Complete Analysis" from both the training and validation datasets as they were absent in the test data. Afterward, we analyzed to determine the distribution of class labels 0 and 1, revealing a notable class imbalance, with the number of instances belonging to class 0 nearly three times higher than those of class 1.

To address this issue, various approaches can be employed. In this study, we opted to mitigate the class imbalance using the focal loss function as our loss function. Our investigation demonstrates the efficacy of focal loss in rectifying class imbalance, enhancing the performance of classes with limited training samples, offering adaptability in adjusting the learning process, and attenuating the impact of noisy data.

3.2 Model

3.2.1 Pre-trained Models

In this study, we tackled the challenge of dealing with long sets of data. To overcome this, we looked into using two models designed to handle large inputs: the Longformer and Big Bird models. These models can handle up to 4096 tokens, which is much more than the BERT model. We also used two pre-trained models specifically trained for legal data: Legal-RoBERTa and Legal-XLM-RoBERTa. These models, like Legal-BERT (Chalkidis et al., 2020), were trained on various legal documents and cases.

Our aim is straightforward: to compare the performance of these models and understand how using pre-trained models with larger input sizes impacts their effectiveness. Through this analysis, we aim to gain insights into the optimal approaches for managing complex legal datasets.

To address the issue of unbalanced data, we implemented the focal loss function, a method that has shown promising outcomes in previous research (Lin et al., 2018) (Wang et al., 2022). The Focal Loss function formally incorporates a factor of $(1 - p_t)^\gamma$ into the standard cross-entropy criterion. This adjustment diminishes the relative loss for accurately classified examples ($p_t > 0.5$), thereby intensifying the focus on challenging instances that

are misclassified.

$$CE(p_t) = -log(p_t) \quad (1)$$

$$FL(p_t) = -(1-p_t)^{\gamma} log(p_t) \quad (2)$$

There is a tunable focusing parameter $\gamma \geq 0$. Figure 1 illustrates the varied impact of this parameter across a range of values.

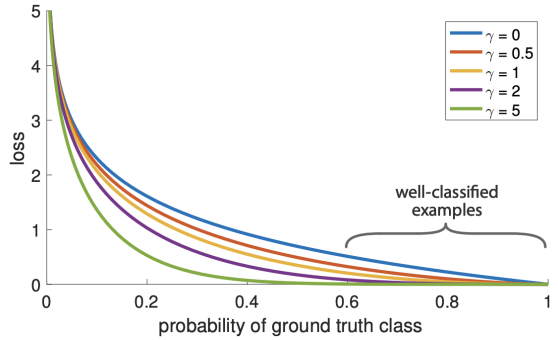


Figure 1: Comparison between cross entropy and focal loss

3.2.2 Large Language Models

In addition, we leveraged three popular large language models to assess their performance within the given task: OpenAI's GPT 3.5 model, Google's Gemini model, and Bing's Copilot model. Later, we formulated the task in the form of prompts, refined them through prompt engineering techniques, and presented them to the large language models. Our objective was to extract the desired answer (0 or 1) from the models' responses, thereby assessing their performance and capabilities in handling the task. We conducted extensive tests on numerous prompts to identify the most effective ones. Employing prompt engineering methodologies alongside large language models, we refined these prompts to enhance their performance and effectiveness in achieving our objectives.

4 Experimental setup

We utilized a dataset of 848 data points, dividing it into 80% for the training dataset, 10% for the validation dataset, and 10% for the test dataset. Our experimental approach involved exploring three main inquiries:

1. Exploration of Longformer and Big Bird models, tailored to effectively process lengthy data inputs.

2. Utilization of pre-trained models for legal contexts, including Legal-RoBERTa and Legal-XLM-RoBERTa, to ascertain their efficacy in legal text analysis.

Hyperparameter	Value
Optimizer	AdamW
Learning Rate	5e-5
Epochs	3
Batch Size	1
Loss Function	Focal Loss

Table 1: Hyperparameter values

Model	Accuracy	$\mathbf{F_1}$
Longformer	**0.79**	0.44
Big Bird	**0.79**	0.44
Legal-BERT	0.75	**0.58**
Legal-RoBERTa	**0.79**	0.44
Legal-XLM-RoBERTa	0.78	0.43

Table 2: Accuracy and macro F1-score of first and second parts' models on the validation set

3. Implementation of the Zero-shot methodology with prompt feeding across three distinct models: GPT 3.5[1], Gemini[2], and Copilot[3], aimed at exploring their adaptability and performance across diverse tasks.

In the initial phase, we employed the AdamW optimizer function with a learning rate set at 5e-5 for both models, conducting training over 3 epochs. The specific values chosen for each hyperparameter are listed in Table 1.

Then, in the second phase, we adapted the original code from the article with the necessary modifications.[4] The original code featured the utilization of Sliding Window Simple (SWS) and Sliding Window Complex (SWC) methods with the Legal-BERT model. We made adjustments to certain sections of the code to ensure compatibility with the test input data. Throughout our evaluation, we utilized the macro F_1 score as the primary evaluation metric.

In the final phase of our experiments, we used the API keys provided by OpenAI and Google. Unfortunately, despite our efforts, we encountered challenges locating the official API for Bing. Given its superior accuracy compared to other models, we decided to manually record the results of the test dataset. Throughout this phase, we employed a variety of prompts and iteratively refined them.

[1] https://chat.openai.com/
[2] https://gemini.google.com/app
[3] https://www.bing.com/chat
[4] https://github.com/trusthlt/legal-argument-reasoning-task

Model	Accuracy	F_1
GPT 3.5	**0.69**	0.59
Gemini	0.44	0.44
Copilot	0.67	**0.64**

Table 3: Macro F1-score of large language models on the test set

Our investigation revealed that incorporating terms such as "step by step" or asking for explanations of the inference steps to achieve the desired outcome positively impacted the model's performance. Furthermore, we encountered limitations in prompt completeness due to the token size constraints inherent in the input models. For instance, Bing's Copilot supported a maximum input size of 4000 characters, which proved insufficient for processing our long samples. Here is an example of an input prompt:

I will provide a question, an answer, and an explanation. Your task is to determine if the answer is correct based on the explanation provided. After reading the explanation, please respond with 'yes' if the answer is correct, or 'no' if it is incorrect.

Question: {question}
Proposed Answer: {answer}
Explanation: {explanation}

Is the proposed answer correct based on the explanation? (yes or no)
Please provide your detailed reason for your choice.
Then, reevaluate and check whether the selected answer is logical or not.
Please use the following format:
<selected_answer>: yes/no
<reason>: your reason for the initial choice
<reason for logical check>: your reason for reevaluation

5 Results

After completing the aforementioned three phases, our investigation revealed that despite fine-tuning, existing models struggled to effectively analyze lengthy data within challenging legal contexts, encountering training process issues and yielding suboptimal outputs. Moreover, the most promising result emerged from fine-tuning the Legal-BERT model, serving as the baseline. While we continue to analyze the underlying reasons for this outcome, initial observations suggest that the learning challenge may be linked to the specific characteristics of the dataset employed. Table 2 presents the performance metrics of the models from the initial two phases, evaluated on the validation dataset.

When it comes to evaluating the results of the zero-shot method, we identified its considerable potential and Bing's Copilot model emerged as the top performer, surpassing expectations. Following suit, the GPT 3.5 model presented moderate performance, while the Gemini model fell short of expected levels. The success of the Copilot model lies in its ability to address previous challenges associated with GPT models by leveraging real-time information accessible through the internet. Table 3 presents the results achieved from employing this method on the test dataset, representing the unofficial results submitted during the post-evaluation phase.

6 Conclusion

In this paper, we present methods designed for classifying lengthy legal cases. We divided our exploration into three main parts:

Firstly, we looked into models with large input token sizes such as Longformer and Big Bird. Secondly, we examined pre-trained models specifically fine-tuned for legal data, such as Legal-RoBERTa and Legal-XLM-RoBERTa. Lastly, we tested the zero-shot method across three major language models.

Among these methods, we found that the zero-shot technique and Bing's Copilot model showed the most promising performance. As for future works, we can explore techniques like data summarization, collaborative approaches such as the round table technique, trying various hyperparameters, and refine prompts to further enhance model performance. These efforts have the potential to advance the effectiveness of classification tasks in legal contexts.

As a future work, it would be valuable to explore additional large language models. These models offer extensive capabilities, especially in summarizing lengthy datasets, which could help evaluate various models' performance. However, it is necessary to note that during summarization, some important details might be overlooked.

Another avenue for future research involves testing the effectiveness of a multi-model approach.

(Chen et al., 2023) This method entails bringing together different large language model agents in a round table conference format. This setup encourages diverse perspectives and discussions to foster consensus. By adopting this approach, researchers can tap into the combined intelligence of multiple models, potentially enriching analysis across various tasks and domains. In this competition, our team achieved the 17th rank out of 21 groups. Our only submission during the evaluation phase utilized the basic prompt and the GPT model. However, significant improvements were made during the post-evaluation phase, resulting in a much higher level of accuracy.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The Legal Argument Reasoning Task in Civil Procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, UAE. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school.

Ilias Chalkidis*, Nicolas Garneau*, Catalina Goanta, Daniel Martin Katz, and Anders Søgaard. 2023. LeXFiles and LegalLAMA: Facilitating English Multinational Legal Language Model Development. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023. Reconcile: Round-table conference improves reasoning via consensus among diverse llms.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. Saullm-7b: A pioneering large language model for law.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023a. Chatlaw: Open-source legal large language model with integrated external knowledge bases.

Junyun Cui, Xiaoyu Shen, and Shaochun Wen. 2023b. A survey on legal judgment prediction: Datasets, metrics, models and challenges. *IEEE Access*, 11:102050–102071.

J.W. Glannon. 2018. *Glannon Guide to Civil Procedure: Learning Civil Procedure Through Multiple-Choice Questions and Analysis*. Glannon Guides Series. Aspen Publishing.

Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. Sailer: Structure-aware pre-trained language model for legal case retrieval.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Interpretable long-form legal question answering with retrieval-augmented large language models.

Joel Niklaus, Veton Matoshi, Matthias Sturmer, Ilias Chalkidis, and Daniel E. Ho. 2023. Multilegalpile: A 689gb multilingual legal corpus. *ArXiv*, abs/2306.02069.

Jaromir Savelka, Kevin D. Ashley, Morgan A. Gray, Hannes Westermann, and Huihui Xu. 2023. Explaining legal concepts with augmented large language models (gpt-4).

Cheng Wang, Jorge Balazs, György Szarvas, Patrick Ernst, Lahari Poddar, and Pavel Danchenko. 2022. Calibrating imbalanced classifiers with focal loss: An empirical study. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 145–153, Abu Dhabi, UAE. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

TrustAI at SemEval-2024 Task 8: A Comprehensive Analysis of Multi-domain Machine Generated Text Detection Techniques

Ashok Urlana **Aditya Saibewar** **Bala Mallikarjunarao Garlapati**
Charaka Vinayak Kumar **Ajeet Kumar Singh** **Srinivasa Rao Chalamala**
TCS Research, Hyderabad, India
ashok.urlana@tcs.com, aditya.saibewar@tcs.com, balamallikarjuna.g@tcs.com
charaka.v@tcs.com, ajeetk.singh1@tcs.com, chalamala.srao@tcs.com

Abstract

The Large Language Models (LLMs) exhibit remarkable ability to generate fluent content across a wide spectrum of user queries. However, this capability has raised concerns regarding misinformation and personal information leakage. In this paper, we present our methods for the SemEval2024 Task8, aiming to detect machine-generated text across various domains in both mono-lingual and multi-lingual contexts. Our study comprehensively analyzes various methods to detect machine-generated text, including statistical, neural, and pre-trained model approaches. We also detail our experimental setup and perform a in-depth error analysis to evaluate the effectiveness of these methods. Our methods obtain an accuracy of 86.9% on the test set of subtask-A mono and 83.7% for subtask-B. Furthermore, we also highlight the challenges and essential factors for consideration in future studies.

1 Introduction

Recent advancements in Large Language Models (LLMs) have facilitated a wide range of applications, notably in content generation (Chung et al., 2023). While LLMs offer creative and informative content generation capabilities, concerns such as misinformation, fake news, personal information leakage, legal and ethical issues have emerged (Chen and Shu, 2023; Li, 2023; Kim et al., 2023). Consequently, detecting machine-generated text has become a crucial task to address these aforementioned challenges.

The identification of machine-generated text is still an open challenge because of its overlapping similarities with human-written text. The current text generation models produce text that is strikingly similar to human language in terms of grammaticality, coherency, fluency, and utilization of real-world knowledge (Radford et al., 2019; Zellers et al., 2019; Brown et al., 2020). However, variations in sentence length, the presence of noisy data, and the generation of incomplete sentences are common indicators of machine-generated text.

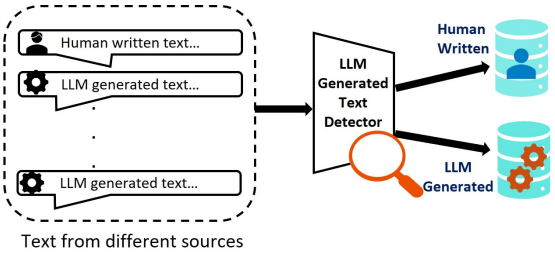


Figure 1: Block diagram for machine-generated text detection.

1.1 Essence of LLM generated text detection

LLMs' open-ended text generation techniques have sparked various concerns across domains (Jo et al., 2023). It has been demonstrated that LLMs have the potential to generate misinformation and fake news (Chen and Shu, 2023), which can be catastrophic in healthcare (Zhou et al., 2023), public safety, education, and finance. Moreover, LLMs can generate text without source attribution, raising the risk of plagiarism (Quidwai et al., 2023), and can include legal and ethical concerns too (Li, 2023).

Furthermore, when LLMs are used in enterprise applications there can be concerns of intellectual property rights infringement (Zhao et al., 2024) such as generated content might contain trademarks or branding elements (Ren et al., 2024). Lastly, LLMs can aggravate security concerns by generating phishing emails (Bethany et al., 2024), fake reviews (Adelani et al., 2020), hallucinations (Huang et al., 2023), biased content (Fang et al., 2023; Dai et al., 2024), and personal information leakage (Kim et al., 2023).

1.2 Tasks

The main objective of the competition is to differentiate text based on the source of its generation

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 927–934

method (see Figure 1), with specific importance given to machine-generated text and human-written texts (Wang et al., 2024a). The competition consists of three tasks Subtask A, Subtask B, and Subtask C. Our study focuses on Subtasks A and B.

Subtask A. Binary Human-Written vs. Machine-Generated Text Classification: This task aims to distinguish between human-written or machine-generated text. This task acts as a binary classification. Subtask A is again subdivided into the following two categories. *Mono-lingual:* The text is in the English language. *Multi-lingual:* The text is in English, Chinese, Russian, Urdu, Indonesian, Arabic, and Bulgarian languages.

Subtask B. Multi-Way Machine-Generated Text Classification: This task aims to classify the given text into six distinct classes, which are 'human', 'chatGPT', 'cohere', 'davinci', 'bloomz', 'dolly' with each class representing the source of its generation. This task acts as a multi-class classification.

The key contributions of this work include, 1) We present a comprehensive analysis of various machine-generated text detection techniques for multi-domain mono and multi-lingual data, 2) We provide a detailed experimental setup for statistical, neural, and pre-trained models along with corresponding error analysis, 3) We emphasize the discussions and future perspectives derived from the findings of the study.

2 Related Work

Recent works on LLM-generated[1] text detection has shown promising results. Statistical methods are used to detect the LLM-generated text by utilizing the entropy (Shen et al., 2023), and N-gram frequency (Tassopoulou et al., 2021). Some other studies uses the fact that language models assign high probability for the repeated sentences which is often AI model generated and ranks the AI model generated sentence Krishna et al. (2022). In a study, OpenAI has trained a classifier to detect LLM-generated text using the RoBERTa-based model (Solaiman et al., 2019).

Some of the widely-used methods adopted the GPT detectors such as OpenAI detection classifier[2], GPTZero[3], and ZeroGPT[4]. Another variant is DetectGPT (Mitchell et al., 2023), which works on the assumption of LLM-generated text lies in the negative curvature region of the log-likelihood. Using this approach, DetectGPT perturbs the input text using masked language models, such as BERT (Devlin et al., 2018), BART (Lewis et al., 2019), T5 (Raffel et al., 2019) and compare the log probability of the text and masked filled variants. Similarly, few works utilized the different decoding strategies including top-k, nucleus, and temperature sampling to generate the text from GPT2 and BERT based models employed to perform binary classification to label text as human-written or machine generated (Ippolito et al., 2020).

Recently, watermarking methods have been used in enterprises to protect the intellectual properties and fair use of the generation models. However these techniques simplify the detection of the LLM-generated output text by synonym replacement over generated outputs and text level posthoc lexical substitutions (Li et al., 2023; Sadasivan et al., 2023), and soft watermarking was introduced in (Kirchenbauer et al., 2023) using green and red token lists. Hidden space operations were also introduced by injecting secret signals into the probability vector of each target token (Zhao et al., 2023).
Bhattacharjee and Liu (2023) proposed a method which triggers when the text has common words randomly assembled as it is easier to find than identifying unique and rare tokens. Sadasivan et al. (2023) focused on zero-shot AI text detection by using two clusters depending on watermarked or not. Another study (Wang et al., 2024c), proposed a benchmark framework consists of an input module, a detection module and an evaluation module for machine generated text detection against human-written text. In contrast to existing works, this study presents the multi-domain multi-lingual machine generated text detection techniques.

3 Datasets

This section given an overview of the dataset utilized and the corresponding analysis.

3.1 Source and acquisition

The task organizers provided the dataset[5] for all the tasks (§1.2). The dataset is an extension of the M4 dataset (Wang et al., 2024b). The dataset provided for this task consists of machine-generated text and human-written text. The human-written text is gathered from various sources such as

[1]We interchangeably use the terms 'LLM-generated' or 'machine-generated'

[2]https://platform.openai.com/ai-text-classifier

[3]https://gptzero.me/

[4]https://www.zerogpt.com/

[5]https://github.com/mbzuai-nlp/SemEval2024-task8

	Subtask - A (Mono-lingual)			Subtask - A (Multi-lingual)			Subtask - B		
	Train	Development	Test	Train	Development	Test	Train	Development	Test
# Samples	**119757**	**5000**	**34272**	**172417**	**4000**	**42378**	**71027**	**3000**	**18000**
# Avg sentences	23	17	18	19	10	17	18	12	18
# Minimum sentences	1	1	1	0	1	1	1	1	1
# Maximum sentences	1583	699	882	1583	59	882	699	477	882
# Median sentences	14	9	18	12	10	17	12	10	17
# Avg words	530	394	437	445	222	396	398	267	414
# Minimum words	2	7	12	0	41	12	6	7	12
# Maximum words	38070	19115	2946	38070	2081	6308	19115	1484	2946
# Median words	319	213	424	296	218	379	290	217	413

Table 1: SemEval 2024 Task 8 data statistics.

Wikipedia, WikiHow (Koupaee and Wang, 2018), arXiv, and PeerRead (Kang et al., 2018), Reddit (Fan et al., 2019) for English, Baike and Web question answering (QA) for Chinese, news for Urdu, news for Indonesian and RuATD (Shamardina et al., 2022) for Russian. On the other hand, the machine-generated text is gathered by prompting different multi-lingual LLMs: ChatGPT (Achiam et al., 2023), BLOOMz (Muennighoff et al., 2023), textdavinci-003, FlanT5 (Chung et al., 2022), Cohere, Dolly-v2, and LLaMa (Touvron et al., 2023).

3.2 Exploratory data analysis

Preliminary analysis of data is a crucial step that is required to understand the dataset characteristics. We have observed that the number of sentences in each task data varies from 1 to a few hundred. Particularly, a few samples in the multi-lingual training data consist of empty samples as well. Another point to note is, that the number of sentences in the multi-lingual train and development varies a lot, which indicates the dataset obtained from different sources. There are a few cases, where some of the samples consist of more than 38k tokens in a single sample. With these observations, to experiment on cleaned data, we employ two types of pre-processing settings. The former (Version-1) applies heuristic-based pre-processing and sub-word removal, whereas the latter (Version-2) applies only heuristic-based pre-processing. We reported the detailed analysis of the dataset statistics in Table 1.

4 System Overview

This section offers various approaches employed to perform machine-generated text identification. Our approaches are categorized into 1) statistical, 2) neural, and 3) pre-trained models.

4.1 Methodology

4.1.1 Statistical methods

To understand the effectiveness of statistical models, we experimented with a wide range of statistical models and their variants including ensemble approaches. The statistical models including Logistic Regression (LR), SVM, MLP, LightGBM and some of the ensemble models detailed in Table 3.

4.1.2 Neural methods

Neural networks have demonstrated remarkable success in various domains, from image and speech recognition to natural language processing. We experiment with Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and their combinations. We utilize FastText [(Joulin et al., 2016), (Bojanowski et al., 2017)] embeddings to capture hierarchical patterns within the text data.

4.1.3 Pre-trained models

Self-supervised pre-trained models have been effective for the classification tasks. In this study, we experiment with a wide range of pre-trained models trained on either open-source or language model-generated data. The pretrained models including BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), DistilRoBERTa base (Sanh et al., 2019), RoBERTa Base OpenAI Detector (Solaiman et al., 2019), XLM RoBERTa (Conneau et al., 2019).

4.2 Experimental setup

For all the experiments, we have utilized the default data splits provided by the task organizers. For all the statistical models, four types of embeddings were employed namely counter vectors,

Models	Subtask - A (Monolingual)				Subtask - A (Multilingual)				Subtask - B			
	Count	Word	N-gram	Character	Count	Word	N-gram	Character	Count	Word	N-gram	Character
LR	0.544	0.566	**0.712**	0.615	0.511	0.516	0.498	0.561	0.544	0.514	0.519	0.558
Naive Bayes	0.506	0.520	0.568	0.599	0.510	0.515	0.489	0.509	0.463	0.533	0.495	0.354
SVM	0.534	0.573	0.708	0.634	0.344	0.494	0.512	0.571	0.569	0.550	0.518	0.573
Random Forest	0.576	0.614	0.619	0.682	0.465	0.517	0.504	0.559	0.579	0.462	0.429	0.408
XG Boost	0.584	0.623	0.639	-	0.499	0.507	0.558	-	0.605	**0.619**	0.591	-
MLP	0.594	0.604	0.683	0.647	0.544	0.528	0.485	**0.609**	0.529	0.506	0.493	0.583

Table 2: Accuracy of statistical models development set; LR refers to Logistic Regression, Subtask-B deals with multi-class classification task.

Model	Subtask-A (Monolingual)	Subtask-B
Naive Bayes + SGDClassifier + LightGBM	0.714	0.708

Table 3: Ensemble model Accuracy scores on development set.

Model	Subtask-A Mono	Subtask-A Multi	Subtask-B
CNN + FastText	0.711	0.545	0.652
RNN + LSTM + FastText	0.682	0.615	0.549
Bidirectional RNN + FastText	0.689	0.579	0.582

Table 4: Accuracy of neural models on development set.

Task	Model	Accuracy
Subtask-A (Mono)	BERT Base	**0.825**
	BERT Base_v1	0.807
	BERT Base_v2	0.813
	BERT Base_v2	0.809
	RoBERTa Base OpenAI Detector	0.766
Subtask-A (Multi)	BERT Multilingual Base_v2	0.622
	XLM-RoBERTa	**0.766**
	BERT Multilingual Base	0.622
Subtask-B	RoBERTa Large	0.751
	RoBERTa Base OpenAI Detector	**0.753**
	DistilRoBERTa Base	0.733

Table 5: Pre-trained models Accuracy scores on development set; Where v1 and v2 indicates different pre-processing strategies.

word, n-gram, character-level TF-IDF vectors and spaCy embeddings. Moreover, we used the default configurations mentioned in the scikit-learn[6]. Whereas for pre-trained models the list of hyper-parameters details are listed in Table 6. We have not performed any hyperparameter-tuning for our experiments. We conduct most of our experiments using four Nvidia GeForce RTX 2080 Ti (11GB) GPUs. To evaluate all the models, we reported the 'Accuracy' scores.

5 Results and Analysis

This section provides a detailed analysis of the models utilized for subtasks A and B. Our experiments aim to showcase the effectiveness of several machine-generated text detection techniques.

5.1 Subtask A Mono-lingual

We experiment with the statistical and neural models to perform subtasks A and B. All the statistical and ensemble models experimental results on development data are mentioned in Table 2 and Table 3. The results on test data mentioned in Table 7. In the case of statistical models, Logistic Regression obtains the superior performance of 71.2% accuracy using n-gram level TF-IDF embeddings compared to other methods on the development dataset. Whereas in the case of the performance of the test set, our ensemble surpass all the remaining models. We built the ensemble model by creating a custom tokenizer by combining spaCy embedding and TF-IDF with n-gram level range of (3-5) embedding. Moreover, we trained an ensemble model with Naive Bayes, SGDClassifier[7], and LightGBM models which gave 86.9% accuracy on the test set. We experiment with a few neural models with fastText embeddings and out of them CNN+fastText outperforms the other models. We have listed results in Table 4. Moving ahead, we fine-tuned transformer-based pre-trained language models like RoBERTa Base OpenAI detector (Solaiman et al., 2019), which gave 76.6% accuracy on the development set and 78.7% accuracy on test set, BERT base model which gave 82.5% accuracy on the development set and 71.7 % accuracy on test set. The results are detailed in Table 5. Furthermore, we use the pre-processing steps discussed in Section 3.2.

[6] https://scikit-learn.org/stable/supervised_learning.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

Model	Batch size	Epochs	Vocab size
BERT Base	16	10	30522
OpenAI Detector	16	10	50265
BERT Multilingual Base	8	3	30522
XLM-RoBERTa	8	5	250002
RoBERTa Large	4	2	50265
DistilRoBERTa Base	16	10	29409

Table 6: Experimental setup for pre-trained models. For all the models max source length set to 512 and learning rate $5e^{-5}$.

Fine-tuned the BERT base model with version-1's pre-processed data gave 80.7% on the development dataset and 71.7% on the test set. Then we fine-tuned the BERT base model with version-2 pre-processed data gave 81.3% on the development dataset and 69.7% on the test set. We secured 24^{th} rank out of 137 participants.

We observed that statistical models that performed modestly on the development set generalized effectively to the test set, whereas some pre-trained language models, despite performing well on the development set, struggled to generalize on test set. This discrepancy may stem from the differing sources of the training and development sets ('arxiv', 'reddit', 'wikihow', 'wikipedia', 'peerread') compared to the test set, potentially causing over-fitting of the pre-trained models on the training data and hindering their performance on the test set.

5.2 Subtask A Multi-lingual

For subtask A multi-lingual, we fine-tuned BERT Multilingual Base and XLM RoBERTa base models. BERT Multilingual Base along with version-2 pre-processed data resulted in 62.2% accuracy on the development set and 73.8% accuracy on the test set. Moreover, despite the decent performance of XLM-RoBERTa on the development set with 76.6% accuracy, the performance of on test set is sub-par. Furthermore, the BERT Multilingual base gave 62.2% accuracy on the development set and 73.1% accuracy on the test set. As mentioned in Section 3.2, we observed that, the multi-lingual data consists of empty samples. Hence, we fine-tuned the BERT Multilingual Base model on the version-2 of the pre-processed data, which helped in improving the accuracy of the test set even if we had the same accuracy on development set.

5.3 Subtask B

Subtask B deals with multi-class classification task. For this task, we have conducted experiments using the statistical models as well as the pre-trained language models. MLP model gave the best accuracy on the development set with 60.9% accuracy. Our ensemble approach obtains 70.8% accuracy on the development set and 65% accuracy on the test set. Moreover, we experimented with RoBERTa Base OpenAI Detector gave 75.3% on the development set and 83.7% accuracy on the test set. Whereas, the DistilRoBERTa base obtains 73.3% accuracy on the development set and 79.1% accuracy on the test set and secured 17^{th} rank out of 86 participants.

Task	Model	Accuracy
Subtask - A (Mono)	Baseline	0.74
	Naive bayes + SGDClassifier + LightGBM*	**0.869**
	RoBERTa Base OpenAI Detector	0.787
	BERT Base_v1	0.717
	BERT Base	0.715
	BERT Base_v2	0.697
Subtask - A (Multi)	Baseline	0.72
	BERT Multilingual Base_v2	**0.738**
	BERT Multilingual Base	0.731
	XLM-RoBERTa *	0.50
Subtask - B	Baseline	0.75
	RoBERTa Base OpenAI Detector	**0.837**
	DistilRoBERTa Base*	0.791
	Naive bayes + SGDClassifier+ LightGBM	0.650

Table 7: Test set accuracy results; *entries are the official submission models of the competition.

6 Conclusions

The study explores different methodologies for detecting machine-generation text, leveraging statistical, neural, and pre-trained models. We observe that the ensemble models are more effective in classifying the mono-lingual data (Subtask-A mono), while models trained on GPT2-text surpass other models in multi-class classification.

7 Limitations

In our study, due to computational constraints, we have not performed experiments with any large language models. Current evaluation has been limited to conventional ML and pre-tained language models. Some of our experimental methods perform better on development data, where as there is a significant drop on test data, this may result in lack of generalization.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *Advanced Information Networking and Applications*, pages 1341–1354, Cham. Springer International Publishing.

Mazal Bethany, Athanasios Galiopoulos, Emet Bethany, Mohammad Bahrami Karkevandi, Nishant Vishwamitra, and Peyman Najafirad. 2024. Large language model lateral spear phishing: A comparative study in large-scale organizational settings.

Amrita Bhattacharjee and Huan Liu. 2023. Fighting fire with fire: Can chatgpt detect ai-generated text?

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Canyu Chen and Kai Shu. 2023. Combating misinformation in the age of llms: Opportunities and challenges.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, Gang Wang, and Jun Xu. 2024. Llms may dominate information access: Neural retrievers are biased towards llm-generated texts.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Xiao Fang, Shangkun Che, Minjia Mao, Hongzhe Zhang, Ming Zhao, and Xiaohang Zhao. 2023. Bias of ai-generated content: An examination of news produced by large language models.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Eunkyung Jo, Daniel A Epstein, Hyunhoon Jung, and Young-Ho Kim. 2023. Understanding the benefits and challenges of deploying conversational ai leveraging large language models for public health intervention. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–16.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. Propile: Probing privacy leakage in large language models.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305.

Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. RankGen: Improving text generation with large ranking models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 199–232, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Zihao Li. 2023. The dark side of chatgpt: Legal and ethical challenges from stochastic parrots and hallucination.

Zongjie Li, Chaozheng Wang, Shuai Wang, and Cuiyun Gao. 2023. Protecting intellectual property of large language model-based code generation apis via watermarks. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 2336–2350, New York, NY, USA. Association for Computing Machinery.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Ali Quidwai, Chunhui Li, and Parijat Dube. 2023. Beyond black box AI generated plagiarism detection: From sentence to document level. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 727–735, Toronto, Canada. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Hui Liu, Yi Chang, and Jiliang Tang. 2024. Copyright protection in generative ai: A technical perspective.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected?

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the the ruatd shared task 2022 on artificial text detection in russian. In *Computational Linguistics and Intellectual Technologies*. RSUH.

Lujia Shen, Xuhong Zhang, Shouling Ji, Yuwen Pu, Chunpeng Ge, Xing Yang, and Yanghe Feng. 2023. Textdefense: Adversarial text detection based on word importance entropy.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

V. Tassopoulou, G. Retsinas, and P. Maragos. 2021. Enhancing handwritten text recognition with n-gram sequence decomposition and multitask learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10555–10560, Los Alamitos, CA, USA. IEEE Computer Society.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Akim Tsvigun, Jinyan Su, Artem Shelmanov, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024c. MG-Bench: Evaluation benchmark for black-box machine-generated text detection.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Jiawei Zhao, Kejiang Chen, Xiaojian Yuan, Yuang Qi, Weiming Zhang, and Nenghai Yu. 2024. Silent guardian: Protecting text from malicious exploitation by large language models.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023. Protecting language generation models via invisible watermarking. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Pinealai at SemEval-2024 Task 1: Exploring Semantic Relatedness Prediction using Syntactic, TF-IDF, and Distance-Based Features.

Anvi Alex Eponon[1], **Luis Ramos**[1],
Ildar Batyrshin[1], **Grigori Sidorov**[1], **Olga Kolesnikova**[1], **Hiram Calvo**[1]
[1]Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC),
Mexico City, Mexico
{aeponon2023, lramos2020, ibatyr1, sidorov, kolesnikova, hcalvo}@cic.ipn.mx

Abstract

The central aim of this experiment is to establish a system proficient in predicting semantic relatedness between pairs of English texts. Additionally, the study seeks to delve into diverse features capable of enhancing the ability of models to identify semantic relatedness within given sentences. Several strategies have been used that combine TF-IDF, syntactic features, and similarity measures to train machine learning models to predict semantic relatedness between pairs of sentences. The results obtained were above the baseline with an approximate Spearman score of 0.84.

1 Introduction

The prediction of semantic relatedness between texts is a crucial task with applications in various natural language processing domains. In this study, our focus is on creating a system capable of predicting semantic relatedness across languages while investigating the features that contribute to this prediction. The development of such a system is not only beneficial for understanding semantic relationships within texts but also holds promise for enhancing deep learning models in tasks such as assessing sentence representation methods, question answering, and text summarization (Abdalla et al., 2023).

Despite the advancements in word representation techniques, especially using embeddings, the complexity of human languages presents persistent challenges in accurately capturing semantic relatedness. Our experiment primarily concentrates on the English language, acknowledging its significance as a widely used language in various applications. The inherent difficulty in identifying and quantifying the shared elements between two texts necessitates a thoughtful exploration of diverse features and methodologies.

In the subsequent sections, we describe the dataset used for this experiment, outline the methodology employed for predicting semantic relatedness, discuss the results obtained, and conclude with insights into the implications and potential future directions of this research. The research will delve into exploring features that enhance the prediction of semantic textual relatedness by developing several strategies concerning feature extractions and model training.

2 Literature Review

The complexity of machine-based human language modeling involves a nuanced understanding of various linguistic aspects, notably Pragmatics and Semantics (Abdalla et al., 2021; Miller, 1995). This research specifically emphasizes semantic modeling, with a focus on semantic relatedness, as opposed to the more commonly studied word similarity (Islam et al., 2012; Atoum and Otoom, 2016; Yum et al., 2021).

Traditionally, approaches like Bag of Words have been explored (Islam et al., 2012; Feng F. Jin, 2008), but they often fall short in achieving high performance for semantic relatedness tasks. WordNets models, while prioritized, face limitations in language coverage and comprehensive embedding of semantic relationships (Jordan J. Boyd-Graber, 2005).

A notable contribution by (Gomaa, 2019) introduced a model utilizing multiple similarity features, including cosine similarity and Jaccard. Their multi-layer architecture demonstrated that employing various similarity features collectively yields significantly better results than applying each measure in isolation. However, the approach did not add or consider syntactic features for the enhancement of the semantic prediction on textual data.

Recent advancements in deep learning models exhibit superior semantic similarity and relatedness performance. However, there remains a scarcity of research focusing on the distinctive features between Semantic Textual Similarity (STS) and Se-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 935–939

mantic Textual Relatedness (STR), and how models can better capture the nuances of semantic relatedness between words and sentences (Kolb, 2005).

3 Task Description

The primary objective of this experiment is to create a system that can predict the semantic relatedness between pairs of texts across various languages but also explore the different features that could help models identify semantic relatedness between given sentences. Although the current experiment is focused on English, the development of such a system holds the potential to enhance deep learning models for various tasks, including assessing sentence representation methods, question answering, and text summarization (Abdalla et al., 2023).

4 Data Description

SemEval 2024 Track 1 utilized data provided by organizers, featuring sentence pairs in training, development, and test sets. Each instance is annotated with a score indicating semantic textual relatedness, which ranges from 0 (unrelated) to 1 (related). Table 1 presents statistics about the dataset, while Figure 1 illustrates the distribution of scores, including the counts for scores of 1.0, 0.0, >0.80 and <0.50, in the training and development set.

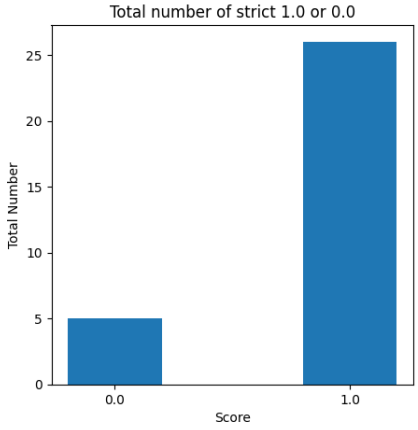
Table 1: Dataset Information

Dataset	Total Pairs	Pairs with Score 1.0	Pairs with Score 0.0
Training	5500	25	5
Development	250	7	123
Test	2600	-	-

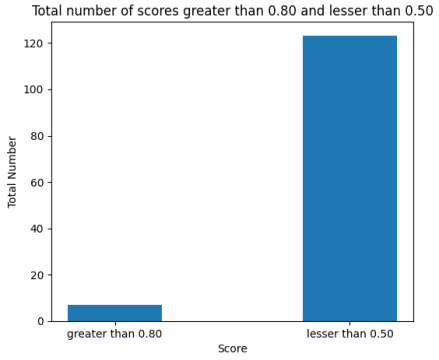
Table 2 displays sample instances, with additional details available in (Abdalla et al., 2021; Ousidhoum et al., 2024b,a).

Instance	Score
It that happens, just pull the plug. if that ever happens, just pull the plug.	1.0
The two little girls jump on the bed. A little girl is jumping down a sandy hill.	0.5
you're taking a sweater in a shop. to taking the life of Conor Greenleaf.	0.03

Table 2: Sample instances of data set



(a) Number of strict 1.0 and 0.0 in the Train set.



(b) Number of scores greater than 0.80 and lesser than 0.5 in the Dev set.

Figure 1: Train and dev scores comparison.

5 Methodology

In this study, our objective is to predict semantic textual relatedness between two texts. We made two key assumptions: firstly, we refrained from preprocessing the corpus to preserve sentence structure, essential for information retrieval and semantic identification (Hirst, 1987).

Secondly, we intentionally excluded Large Language Models (LLMs) from experiments, anticipating challenges in interpreting specific features contributing to semantic identification due to their contextual abilities and complexity (Turton et al., 2020). To extract diverse features, we employed various distance measures, including Jaccard distance, Cosine similarity, Levenshtein distance, and Word Mover's Distance (WMD) (Boubacar, 2014; Kusner et al., 2015; Su et al., 2008). These measures compute the similarity between text pairs based on common words, term vectors, and the minimum distance embedded words need to travel between documents. Feature extraction utilized the

bag-of-words (BoW) technique, specifically Term-Frequency Inverse Document Frequency (TF-IDF) (Hakim et al., 2014), and hidden vectors from a pre-trained SentenceBert model to compute cosine similarity (Reimers and Gurevych, 2019).

Additionally, syntactic features were extracted, parsing each sentence pair to identify words with the same dependency role. This approach resulted in three features: probability of exact word matching, probability of unique words, and probability of related words. The rationale was to explore the impact of words with common dependency roles on predicting semantic relatedness and assess the effect of their absence on the English corpus.

The reason behind using traditional models such as Gradient Boost is that this type of model can easily handle non-linear relationships in texts, which is crucial while capturing semantic relatedness but also can deal with imbalanced datasets (Natekin and Knoll, 2013).

A diagram of our methodology as well as the source code is freely available on GitHub. The link can be found in the Appendix section.

6 Results

The performance metrics of the different models submitted for semantic relatedness are presented. The evaluation metrics include solely the Spearman score. Even though the models submitted were Fasttext and Naive Bayes, during the training phase Linear Regression, XGradient Boost, Random Forest, and an ensemble of two traditional models were trained.

6.1 Train phase

During the training phase, several models were tested using different strategies, and in Table 3 we display the performance for each model. The main strategy that gave the results presented in this document has been explained earlier in the methodology section.

Model	Spearman
Linear Reg.	0.8512
Gradient Boost	**0.8527**
XGB	0.8467
Random Forest	0.8481
Ensemble Model	0.8521

Table 3: Training Model Performances

6.2 Development phase

At this stage of the experiment, the same models were tested on the development dataset which comprised of few number of samples, precisely 250 different instances. In Table 4 we display the performance over the development set.

Model	Spearman
Linear Reg.	0.8512
Gradient Boost	**0.8527**
XGB	0.8467
Random Forest	0.8481
Ensemble Model	0.8521

Table 4: Development Model Performances

6.3 Test phase

At the final stage of the experiments, the Gradient Boost model was chosen for the final tests on the testing dataset which comprised 5000 different instances. In Table 5, we display the results of the Gradient Boost models, compared to the baseline and the highest performance model in the same task.

Team	Spearman	Rank
PALI	0.8595	1
Pinealai	**0.8371**	10
SemRel-Baseline	0.8300	*

Table 5: Final Model Performance Metrics

7 Discussions

In the methodology section, our primary objective is to identify key features that enhance the capability of models in discerning semantic relatedness within textual data. We pursued two distinct approaches. First, we trained various models by employing Tfidf, Jaccard, or extracting specific syntactic features from the texts. This process does not take into account the internal structures of the texts which could give more insights about their meaning.

When exclusively utilizing syntactic features for model training, we achieved a maximum Spearman score of 0.32. Training models solely that used TF-IDF features to compute a cosine similarity and used the metric to predict yielded a separate score of 0.533. Incorporating features extracted from Sbert on top of the syntactic features resulted in a

notable score increase of 2, reaching approximately 0.70. Finally, combining all these strategies during the training phase produced a score of 0.85, with a corresponding score of 0.83 during the testing phase.

8 Conclusion

In conclusion, the study aimed to predict semantic relatedness in English, exploring diverse features and strategies. Limitations included the absence of word sense disambiguation algorithms, the exclusion of Transformer models for explainability, and the decision not to merge training and development sets. Despite these constraints, the models exhibited competitive performance, particularly the Gradient Boost model, which achieved a Spearman score of 0.8371. However, the experiment conducted can not help us derive a conclusion that the model can make a strict difference between semantic relatedness (STR) and semantic textual similarity (STS) in texts. The methodology highlighted the impact of syntactic features, TF-IDF representations, and SentenceBert embeddings. Moving forward, addressing limitations, incorporating advanced algorithms, and leveraging diverse datasets but also developing approaches that help models distinguish between STR and STS will contribute to a deeper understanding of semantic relations in textual data and further improvements in predictive capabilities.

Limitations

The study of semantic relatedness is a vast and tedious endeavor. The research conducted was very limited in many aspects. Firstly, the absence of algorithms specifically targeting direct word sense disambiguation represents a notable limitation. The incorporation of such algorithms could have potentially enhanced the models' effectiveness in this particular task. Also, the research did not explore the preprocessing techniques that could positively impact the semantic relatedness prediction.

Secondly, our study was confined to the training and testing of traditional machine learning models, excluding the exploration of Transformer models. While Transformers might have yielded superior results, their reduced explainability deterred their inclusion in our investigation.

Lastly, the decision not to merge the training and development sets for a final model training phase or add more datasets related to semantics relatedness or even semantic similarity represents another constraint. By solely transitioning to the testing phase with the models having learned solely from the training set given by the organizers, we may have missed opportunities for improvement. Combining both sets or augmenting the datasets through specific techniques in the final training phase could have potentially elevated the models' predictive capabilities, resulting in a more accurate score.

Ethics Statement

We affirm our commitment to ethical research practices and compliance with ACL guidelines in conducting and presenting our study. No ethical concerns or conflicts of interest arose during this research.

Acknowledgements

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20241816, 20241819, and 20240951 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *arXiv preprint arXiv:2110.04845*.

Issa Atoum and Ahmed Fawzi Otoom. 2016. Efficient Hybrid Semantic Text Similarity using Wordnet and a Corpus. *International Journal of Advanced Computer Science and Applications*, 7(9).

Abdoulahi Boubacar. 2014. Valuing semantic relatedness. In *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, pages 1–5. IEEE.

Trevor T. Martin Feng F. Jin, Yiming Y. Zhou. 2008. Sentence similarity based on relevance. *Investigation Group of Applied Mathematics in computing*.

WAEL HASSAN Gomaa. 2019. A multi-layer system for semantic relatedness evaluation. *Journal of Theoretical and Applied Information Technology*, 97(23):3536–3544.

Ari Aulia Hakim, Alva Erwin, Kho I Eng, Maulahikmah Galinium, and Wahyu Muliady. 2014. Automated document classification for news article in bahasa indonesia based on term frequency inverse document frequency (tf-idf) approach. In *2014 6th international conference on information technology and electrical engineering (ICITEE)*, pages 1–4. IEEE.

Graeme Hirst. 1987. *Semantic interpretation and the Resolution of Ambiguity*. Cambridge University Press.

Aminul Islam, Evangelos Milios, and Vlado Kešelj. 2012. 29 *Text similarity using Google tri-grams*.

Daniel N. Osherson Robert E. Shapire Jordan J. Boyd-Graber, Christiane C. Fellbaum. 2005. Adding dense, weighted connections to wordnet. *3rd International Global WordNet Conference, Proceedings*.

Peter Kolb. 2005. Experiments on the difference between semantic similarity and relatedness. *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 81–88.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

George A. Miller. 1995. WordNet. *Communications of The ACM*, 38(11):39–41.

Alexey Natekin and Alois Knoll. 2013. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Zhan Su, Byung-Ryul Ahn, Ki-Yol Eom, Min-Koo Kang, Jin-Pyung Kim, and Moon-Kyun Kim. 2008. Plagiarism detection using the levenshtein distance and smith-waterman algorithm. In *2008 3rd International Conference on Innovative Computing Information and Control*, pages 569–569. IEEE.

Jacob Turton, David R. Vinson, and Robert E. Smith. 2020. Deriving Contextualised Semantic Features from BERT (and Other Transformer Model) Embeddings. *arXiv (Cornell University)*.

Yunjin Yum, Jeong Moon Lee, Moon Joung Jang, Yoojoong Kim, Jong-Ho Kim, Seong-Tae Kim, Unsub Shin, Sanghoun Song, and Hyung Joon Joo. 2021. A word pair dataset for semantic similarity and relatedness in Korean medical vocabulary: reference development and validation. *JMIR medical informatics*, 9(6):e29667.

Appendix

The diagram of our method and the source code can be found on GitHub at this URL: semEval2024 code

Infrrd.ai at SemEval-2024 Task 7: RAG-based end-to-end training to generate headlines and numbers

JiangLong He, Saiteja Tallam, Srirama Nakshathri,
Navaneeth Amarnath, Pratiba KR, Deepak Kumar
Infrrd.ai
{jianglong,saitejatallam,srirama}@infrrd.ai
{navaneethamarnath,pratibakr,deepakumar}@infrrd.ai

Abstract

We propose a training algorithm based on retrieval-augmented generation (RAG) to obtain the most similar training samples. The training samples obtained are used as a reference to perform contextual learning-based fine-tuning of large language models (LLMs). We use the proposed method to generate headlines and extract numerical values from unstructured text. Models are made aware of the presence of numbers in the unstructured text with extended markup language (XML) tags specifically designed to capture the numbers. The headlines of unstructured text are preprocessed to wrap the number and then presented to the model. A number of mathematical operations are also passed as references to cover the chain-of-thought (COT) approach. Therefore, the model can calculate the final value passed to a mathematical operation. We perform the validation of numbers as a post-processing step to verify whether the numerical value calculated by the model is correct or not. The automatic validation of numbers in the generated headline helped the model achieve the best results in human evaluation among the methods involved.

1 Introduction

In our busy lives, we barely have time to read the newspapers or an online article. Even a short period will not be enough to read all the latest news articles from different sources. The headline attached to the article attracts the reader only if it is interesting or provokes interest in the reader. A reader may have different interests and may not cover all areas. Some may be interested in movies, politics, science and technology, economy, environment, governance, sports, celebrities, weather, etc. The information fed to a reader is large and must be condensed and remembered. A unique headline condenses the unstructured text into a few words with some numbers. The numbers are presented to attract the reader's attention from the part of the unstructured text. These numbers can be based on positive or negative sentiments. A negative sentiment has a greater influence than a positive sentiment. A positive sentiment aims to create new information in the reader's mind. However, a negative sentiment harms the reader's mind by correcting and updating the information. Named entities, such as name, location, and number, are easier to remember for a long time than the rest of the text. The named entities are used more often by many people in several contexts with high-frequency usage. The narration in the text is constructed with the entities with a relationship. The occurrence of numbers in the relationship is more frequent than in other entities, especially in news articles.

We perform text summarization on unstructured text to obtain specific and highlighted information. It is helpful in many areas and reduces the time spent on unnecessary or irrelevant texts. Several events would occur in the process from the beginning to the end. All events may not be relevant or may appear as information overload. To reduce the list of events, we are using a text summary. Medical report summary, annual report summary, election results summary, movie reviews, product reviews, and sports reviews will highlight the main results of the research or the results of the conducted activities. A person must read the content to prepare a summary of the text. The likelihood is that many relevant points of the content should be included as part of the summary. Each person can prepare different lists of points using their previous knowledge and preferences. The main points of the different lists will be a central part of any summary. The core part can form a headline to attract readers to read the contents of the unstructured text.

Summary generation is a time-consuming process in which many people must contribute to preparing the highlights of the text content. In natural language processing (NLP), a model has to

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 940–951

process text content sequentially using a seq2seq model and generate these highlights. It reduces the time required to generate highlights, increases knowledge aggregation, and filters interesting content. A vast literature of text summarization tasks in NLP shows the required attributes of a machine. A summary is presented in plain text in most cases and may not always contain a number. In the generation of headlines, we need a few numbers to highlight the content. The numbers in a headline play an important role in attracting readers' attention. Here, the model must process the unstructured text sequentially and locate the numbers. All numbers in the text cannot be part of the headline. Only a few numbers can cover the complete information from the unstructured text. A model has to identify the numbers that cover the news content in order to generate the headline (Cai et al., 2023; Ding et al., 2023; Zhang et al., 2020).

The main contributions of the proposed method are as follows:

- Retrieval-augmented generation (RAG) of training samples to fine-tune a large language model (LLM).
- Chain-of-thought (COT) based generation of mathematical operations by the model.
- Verification of the computed value by the model to increase the confidence of the extracted numbers from the unstructured text.

2 Related Work

Large language models (LLMs) have started to demonstrate reasoning, calculation, knowledge acquisition, planning, and many more (LLAMA2; MISTRAL; OpenAI). At this point, we need to explore the potential capabilities of LLMs by proposing a wide range of problems that deal with a kind of artificial intelligence embedded in the model. In this paper, we study the numerical ability of a model.

EQUATE benchmark was prepared to make quantitative reasoning on different measures in the natural language inference (NLI) (Ravichander et al., 2019). The data set is prepared to understand whether a model can reason on the text. The results show that the models have the ability to reason and obtain an inference for the statements. The scale of the predicted number was done to understand whether a model can find the magnitude of the predicted numbers through an NLP (Chen et al., 2019). A language model does not explicitly distinguish numbers from words (Chen et al., 2023). The notation of a number cannot be clearly understood by the model. This can be due to a missing number in the training data. We cannot provide all numbers to the model by any means, so we need to use different symbols to express the numbers in the text so that the model can use the numbers to perform the reasoning tasks. However, there was scope to add additional challenges to the data set. The NumGLUE was proposed to identify the performance of LLMs through natural language understanding (NLU) (Mishra et al., 2022). There are eight different tasks based on common sense reasoning, arithmetic calculation, quantitative prediction, fill-in-the-blanks, and arithmetic word problems. Natural language optimization (NLOpt) is a competition to solve arithmetic word problems for linear programming problems (Ramamonjison et al., 2023). We proposed an ensemble approach to detect the named entities (NEs) in an unstructured text (He et al., 2022). The solution was generic and detected most of the entities in the text. Of all these tasks, the headline generation or summarization focused on numbers was missing.

The NumHG data set was prepared to cover the headline generation task by focusing on numbers (Chen et al., 2024, 2021; Huang et al., 2023). The NumEval competition is held to evaluate different models that can understand the numbers and generate the headline according to the ground truth specified (NumEval). The model can choose any random number from the text, which may not be relevant in many cases. The model is pushed to perform the calculation that provides the fill-in-the-blank task, where the model has to calculate the missing number from the headline or summary. A model must use mathematical operations to get the answer. The computational ability of the model is explored in this approach and is known as the task of 'numerical reasoning.' Several mathematical operations can be performed using a model. However, the news data set mainly covers the reproduction of the number, the conversion of a word into a number, and the rounding of the number. The distribution of mathematical operations is very narrow. We do not have sufficient samples for other types of mathematical operations, and the model may not be well suited for these types of operations even after fine-tuning. The data set is designed in such a way

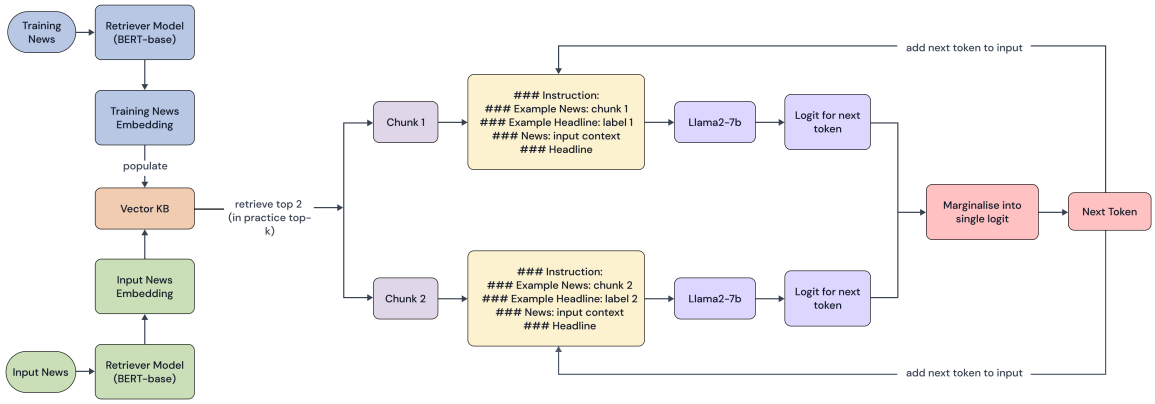


Figure 1: The retrieval-augmented generation (RAG) approach is used in the proposed method to generate headlines and numbers. A vector database is populated using the training news of the NumHG data set, and the training news is retrieved from the database during the inference. The prompted text is as follows: ### Instruction: Generate a Headline for the News and generate Numerical Reasoning for the numbers in the generated headline. Wrap the Numerical Reasoning with XML tags <NR> </NR>. Use the Example News and Example Headline as references. ### News: Input News. ### Headline: .

Table 1: The augmentation of headlines using the chain-of-thought (COT) approach. The NumHG data set provides arithmetic operations in the "calculation" field to be used for numerical reasoning tasks, which are fill-in-the-blanks tasks. The arithmetic operations from the numerical reasoning task are used as a reference to perform augmentation in the ground truth headline.

Ground truth headline	Guy Beat by Police Gets $1K, Lawyers Get $459K
Augmented headline	Guy Beat by Police Gets $1K <NR> paraphrase(1,000,K) </NR>, Lawyers Get $459K <NR> paraphrase(Add(100000,359000),K) </NR>

that the model must be aware of all the numbers presented in the text. Then, the model has to select a few numbers and make the calculation. The numbers are not known, and the mathematical operation is not known to the model in this NumEval task. The model must artificially identify the numbers and choose an appropriate mathematical operation to generate an answer.

3 Proposed method

A given LLM may not have full knowledge of the generation of headlines using a text. We need to provide some support to the model to excel in the task of headline generation. We propose a retrieval-augmented generation (RAG) model that is trained from end to end. The system consists of three modules: Knowledge Base, dense retrieval, and generation modules. Figure 1 shows the block diagram of the proposed approach.

The Knowledge Base (KB) was built using the training data provided. A pre-trained BERT base model is used to encode training news samples into vectors (Devlin et al., 2018). Then, these vectors are indexed using FAISS to enable the task of searching for vector-based similarity (Douze et al., 2024). The same BERT-based model is used during training and inference time. A given input text is encoded in a vector. Encoded vectors are used to search for dense vector similarity. Top-k similar news articles are retrieved by a dense retrieval module. Each result obtained will be added to the original input in a specific template to generate the headline. The purpose is to provide similar examples to generative models to help generate a better result. The prompt used in the experiment is shown in Figure 1. We use the LLAMA2-7b model to generate the output headline (Touvron et al., 2023; LLAMA2). We use the RAG token model (RAG) on the selected Top-k retrieved examples as shown in Figure 1. Each retrieved news is sent along with the input news to obtain two separate token predictions from the LLAMA2-7b model. These token predictions are marginalized, and the process is repeated until all tokens are generated.

Table 2: The ROUGE scores of different models using simple prompt to generate the headlines. The prompt is "Generate a headline for the following passage." The second row in the method name indicates the type of data set and the number of samples in the data set used for evaluation.

Method Name	ROUGE-1	ROUGE-2	ROUGE-L
ChatGPT (gpt-4-1106-preview) (Dry run - 100)	37.61	12.53	32.25
ChatGPT (gpt-4) (Dry run - 100)	36.37	12.25	30.56
ChatGPT (gpt-3.5-turbo) (Dev set - 2365)	35.44	13.16	31.08
LLAMA2-7b (Dev set - 2365)	11.78	4.41	10.37

Table 3: The ROUGE scores of different models using in-context learning approach to generate the headlines. The second row in the method name indicates the type of data set and the number of samples in the data set used for evaluation.

Method Name	ROUGE-1	ROUGE-2	ROUGE-L
Llama-2-13b-chat-hf_results 13b parameter model was used instead of 7b parameter model. The RAG examples were from the dry run set of 100 examples. (Dev set - 2365)	40.98	17.09	36.19
ChatGPT (gpt-3.5-turbo) - BM25 approach The training dataset words are stored in BM25. The search was performed using the development set. (Dev set - 2365)	40.67	17.11	36.15
openbuddy-llama2-70B-v13.2-AWQ_results 70b parameter model was used instead of 7b parameter model. The RAG examples were from the dry run set of 100 examples. (Dev set - 2365)	40.56	16.44	35.90
Mistral-7B-Instruct-v0.2_results Different model with same parameter size is used. The RAG examples were from the dry run set of 100 examples. (Dev set - 2365)	40.48	16.15	35.59
ChatGPT (gpt-3.5-turbo) - RAG examples The training dataset is converted into vectors using Fasttext approach. The similarity search was performed. (Dev set - 2365)	40.41	16.53	35.71
Llama-2-7b-chat-hf_results The RAG examples were from the dry run set of 100 examples (Dev set - 2365)	40.19	16.42	35.62

$$p_{RAG-Token}(y|x) \approx \prod_i^N \sum_{z \in top-k(p(.|x))} p_\eta(z|x) p_\theta(y_i|x, z, y_{1:i-1}) \quad (1)$$

where input sequence x, retrieve documents z, target sequence y. A retriever $p_\eta(z|x)$ with parameter η (BERT-base model) and a generator $p_\theta(y_i|x, z, y_{1:i-1})$ with parameter θ (LLAMA2-7b model). The current token is generated using the previous $i-1$ tokens $y_{1:i-1}$.

3.1 Training

We trained the model from end to end to optimize both the retriever and the generative model together. Here, we aim to train a numerically literate model, which means that the model should be able to understand the numbers in the news and be able to perform mathematical operations on these numbers to arrive at a precise numerical value that will be used in the headline. We encapsulate the numbers in the headline of the news text with XML tags, as shown in the example in Table 1. The tags are an easier way to instruct the model to locate the numbers instead of the model itself looking for the numbers. The content within the XML tags is annotated in the training dataset of the NumHG data set under the numerical reasoning tasks as a "calculation" field. The headline generation task provides a headline as a ground truth. We introduce the information from the numerical reasoning task to the

Table 4: The ROUGE scores of different approaches used to improve the fine-tuned model. The details of number of samples used in the training set. The development set was used for evaluation. The performance improvements with the followed approaches is expressed mnemonically.

Method Name	ROUGE-1	ROUGE-2	ROUGE-L
Dev_Headline_Generation_RagEnd2End + Post_processing The number of training samples used are 14,720. All the numbers in the ground truth headlines are wrapped with XML tags for numerical reasoning task. BRIO generated outputs are used as post- processing step to replace problematic headlines. (Dev set - 2365)	48.08	23.06	43.32
Dev_Headline_Generation_RagEnd2End + XML tags The number of training samples used are 14,720. All the numbers in the ground truth headlines are wrapped with XML tags for numerical reasoning task. (Dev set - 2365)	48.01	23.03	43.32
Dev_Headline_Generation_RagEnd2End + 2nd Inference The number of training samples used are 14,720. All the numbers in the input news are wrapped with XML tags for numerical reasoning task. The second time inference is performed with retrieval using generated headline and news. (Dev set - 2365)	47.56	22.71	43.12
Dev_Headline_Generation_RagEnd2End The number of training samples used are 14,720. All the numbers in the input news are wrapped with XML tags for numerical reasoning task. (Dev set - 2365)	47.46	22.85	43.01
Dev_Headline_Generation_Bart-Large (3 epochs) (Dev set - 2365)	46.40	21.88	41.19
Dev_Headline_Generation_Brio (74 epochs) (Dev set - 2365)	46.08	21.14	40.53

headline generation task. In this respect, we use the "calculation" field in the data set to augment the headline of each training sample. By doing so, we explain the calculation of numbers with the Chain-of-thought (COT) generated in the headline (Wei et al., 2023). Table 1 shows the ground truth headline and the augmented headline.

3.2 Hyperparameters

The model was trained using both the training and the development set for the final submission. The model was trained for 3 epochs on a single A100 GPU with a batch size of 5. The number of documents retrieved is set to 3 for the training period and 5 for the inference period. The learning rate is set to 2e-4, and the linear decay warm-up scheduler is used as a learning rate scheduler with 30 warm-up steps. We use greedy search during inference time to speed up the execution.

We have used LORA (Hu et al., 2021) to fine-tune the LLAMA2-7b model. LORA reduces the number of trainable parameters and memory requirements. The LORA configuration used for the final submission of the headline generation task is as follows: "r": 8, "lora_alpha": 16, "lora_dropout": 0.05, "target_modules": ['gate_proj', 'up_proj', 'o_proj', 'v_proj', 'q_proj', 'k_proj', 'down_proj'].

4 Experiments

LLMs have shown the ability to perform well in unknown tasks even without training. Therefore, it is useful to check whether the model is well suited for the NumEval dataset.

4.1 Out of the box

We have tested several LLMs, and the results are presented in Table 2. Most models perform almost similarly in out-of-the-box scenarios. GPT4 used to take a long time to generate a response. Hence, we experimented on the dry run set provided by the NumEval competition. No examples were provided with inputs to the model. The model must depend on its internal knowledge to generate a headline. The ROUGE scores tabulated above show that the

Table 5: The accuracy of the different methods for the numerical reasoning task. The prompts used for the generation of answer may result in different score.

Method Name	Accuracy (%)
Out-of-the-box LLAMA2-7b	6.53
Out-of-the-box ChatGPT (gpt-3.5-turbo)	43.90
Fine-tuning LLAMA2-7b	86.31
Chain-of-thought (COT)	89.54
COT+under_sampling	82.85
COT+over_sampling	89.00
COT+minority_combined	91.44

model depends on its knowledge to generate a headline. The numerical reasoning task was performed using the fill-in-the-blanks task approach and the out-of-the-box results for LLAMA2 were very low compared to ChatGPT models. The results are summarized in Table 5.

4.2 In-context learning

We use the retrieval-augmented generation (RAG) approach to perform context-based learning. Here, we supplement the input with the retrieved examples. The model must understand the examples shown to generate a headline. The examples provided by the RAG approach are crucial. The number of samples in the knowledge base affects the performance of the model. We have tabulated the ROUGE scores for the RAG-based generation of headlines using different models. There is certainly an improvement from out-of-the-box to context learning. The difference is high in terms of the scores tabulated in Table 3.

4.3 Fine-tuning

The fine-tuning of the LLAMA2-7b model was performed using the RAG method from end to end. The other models were also fine-tuned using different sample sets in the knowledge base (KB). The score was not so much improved compared to LLAMA2-7b. In addition, the BRIO model was trained with different headlines generated by different models from Table 3. The fine-tuned BRIO model also provided performance closer to the best performance method (Liu et al., 2022). However, it was used in the post-processing stage to add the headline if the main model did not perform the mathematical operations correctly. The results of the fine-tuned models are presented in Table 4.

Table 6: The numerical accuracy of different competing methods on the test set.

Position	Team Name	Accuracy Private Leaderboard
1	CTYUN-AI	0.95
2	Zhen Qian	0.94
3	YNU-HPCC	0.94
4	NCL_NLP	0.94
5	NumDecoders	0.91
6	**Infrrd.ai**	**0.90**
7	Hc	0.88
8	NLPFin	0.86
9	NP-Problem	0.86
10	AIRah	0.83
11	Noot Noot	0.77
12	GPT-3.5 (Baseline)	0.74
13	Sina Alinejad	0.74
14	StFX-NLP	0.60

4.4 Chain-of-thought (COT)

The numerical reasoning task requires the collection of the input text to perform the calculation. We provide a series of steps as instructions for the model. Suppose the model has to extract two numbers, say 19 and 16, and then perform addition to calculate the final score. We provide the model with a mathematical operation such as ADD(19,16) in the reasoning steps to get the final answer such as 35. The annotation of the input text to generate the instructions was simple in the NumEval data set. However, the model did not complete the calculations for some samples. Sometimes the model would not provide the final answer. The failure is unknown but based on the last step taken by the model. The last step was completed when the answer was empty. The COT-based reasoning improved the accuracy of the model compared to the LLAMA2-7b fine-tuned model. The results are tabulated in Table 5. The COT approach is superior compared to the fine-tuned model in the numerical reasoning task.

The results are tabulated in tables. 1-4 are based on experiments carried out on the development set. These experiments are conducted to identify the appropriate tools to help improve the performance of the model. The results show that current LLMs may not know completely how to generate a headline for a given text piece. The model needs the support of examples to generate headlines. The model also requires fine-tuning to reproduce answers closer to the ground truth.

Table 7: Automated evaluation of headline generation performed on the results of the test set.

Team Name	Overall	Num Acc. Copy	Reasoning	ROUGE 1	ROUGE 2	ROUGE L	BERTScore P	BERTScore R	BERTScore F1	MoverScore
ClusterCore	38.233	51.571	13.942	33.467	11.837	28.927	31.876	42.232	37.026	56.405
Noot Noot	38.393	57.481	3.6331	31.47	11.139	27.284	25.389	43.977	34.539	55.559
Infrrd.ai	**65.840**	**68.354**	**61.263**	**46.789**	**22.36**	**42.095**	**51.005**	**47.260**	**49.134**	**59.731**
np_problem	73.487	76.908	67.257	39.816	17.577	34.339	27.800	48.557	37.816	57.024
hinoki	62.347	66.284	55.177	43.072	19.719	38.999	47.223	43.444	45.342	58.711
Challenges	72.956	82.170	56.176	31.220	12.235	26.859	19.530	47.559	33.132	55.362
NCL_NLP	62.122	65.536	55.904	43.506	19.388	38.878	46.402	45.039	45.734	58.861
YNU-HPCC	69.044	73.018	61.807	48.852	24.681	44.175	51.553	50.095	50.381	60.551
NoNameTeam	55.715	57.681	52.134	40.646	17.261	35.745	44.256	40.387	42.324	57.736

5 Discussion and Results

The results of the test set based on the proposed method are presented in Tables 6, 7 and 8. The number of samples in the numerical reasoning task is 4921, and the number of samples in the headline generation task is 5227. The same proposed approach is used to estimate the results of the test set. The number of epochs used to train the model for the numerical reasoning task and the headline generation task are 1 and 3, respectively. Since LLMs consist of a large number of parameters, it is very difficult to make any internal changes. We could at least change some blocks that are connected to the LLMs either on the input or the output side. The changes to these blocks will be discussed in this section.

Table 8: Human evaluation of headline generation using reward points awarded by the human evaluator on the selected test set samples.

Team Name	Num Acc. (50 Headlines)	Recommendation (100 News)
ClusterCore	1.60	31
Noot Noot	1.68	11
Infrrd.ai	**1.81**	**22**
np_problem	1.57	14
hinoki	1.67	16
Challenges	1.70	10
NCL_NLP	1.73	16
YNU-HPCC	1.69	15
NoNameTeam	1.59	12

5.1 Numerical reasoning

The basic LLAMA2-7b model used in the out-of-the-box approach did not perform well in the fill-in-the-blank task. However, after fine-tuning the LLAMA2-7b model. The model was able to generate the answer for most of the samples correctly. However, there was still a gap in achieving higher numerical accuracy. The chain-of-thought (COT) approach is used to improve the model's accuracy. The improvement was marginal but observable in terms of numerical accuracy. The chain of thought forced the model to perform numerical reasoning in steps. The model would not complete some of the steps, but the answer was better than the simple fine-tuning approach. When the model fails to complete the last step, the answer is obtained through automated calculation. In addition to COT, we also conducted experiments for minor samples like undersampling and oversampling to train the model. The score was almost the same, and the changes were minimal. The higher the number of parameters (13b) in the model, the better the result than the smaller number of parameters (7b) in the model. Finally, we used the RAG-based approach to complete the fill-in-the-blank task. The combination of RAG and COT improved the model's ability to generate answers more accurately than the other approaches tested during the training period. Tables 5 and 6 show numerical accuracy on the development set and test set, respectively. We performed post-processing on the numerical value generated by the model by filling the empty values through understanding mathematical operations, but the numbers used by the model were not correct in most cases, leading to a wrong value and not improving the numerical accuracy.

5.2 Headline generation

We began to test the ability of LLMs to generate headlines using the out-of-the-box approach. The ROUGE scores were not satisfactory compared to the competition benchmarks (Huang et al., 2023). We used a context-based learning approach with which LLMs could understand the given examples and generate a much better headline. Even then, the ROUGE scores were below the benchmark. We

started fine-tuning the LLMs and got closer to the benchmark. We evaluated various types of models in the development set using all possible combinations. One interesting thing to identify is the numerically aware LLMs. Initially, we placed XML tags in the input text which provided a small improvement in the ROUGE scores in Table 4. We also performed inference for the second time using the first generated headline as part of RAG, which also provided a small improvement in the scores. Finally, we placed XML tags around the numerical values in the headlines to execute the COT approach. LLMs could understand that the answer should be selected from the input text and generate a mathematical operation that can complete the final answer. This gave the best score in the development set. Tables 7 and 8 show automated and human evaluation metrics for the submitted methods. The proposed method is second for most performance measures that are automatically calculated. In human assessment, the proposed method ranks first and second in terms of numerical accuracy and recommended headline, respectively.

5.2.1 Post-processing

We believe that the main contribution to numerical accuracy in Table 8 is verification. Verification is an important step in the process of finalizing the answer, which automatically improves confidence. One of the reasons for the verification is to select the numbers of the input text and compare them in the generated headline. There may be many numbers, but all of them cannot be used in the generation of headlines. Only a few numbers are used to complete the numerical calculation. Sometimes the model does not complete the calculation. We need to verify the steps followed by the model and fix some of the steps to improve the performance of the model in an automated way. The model generates the mathematical operations in an XML tag. The mathematical operations are processed with the numbers, which are part of the operation to verify that the number generated by the model is correct. We provide a few examples without errors in Appendix A.1 and with errors in Appendix A.2 in the verification stage. If the model fails to verify the generated numbers with the actual mathematical operation, a headline generated by the BRIO model is used to replace the headline generated by the main model (Liu et al., 2022). The number of samples with the replaced headlines is less than one percentage of the total number of samples in the test set. The percentage metric aligns with the top-2 reported results for the development set in Table 4. We were unable to fully explore the ensemble approach for LLMs but tried to combine the results of multiple models. A series of simple rules have been used to check and replace the headlines.

We instructed the model to generate a list of headlines, and then a new model was trained using reinforcement learning with human feedback (RLHF) (Böhm et al., 2019). The ROUGE scores from the RLHF-based model were better than a single headline generator but much less than the context-based learning approach. So, we have not reported the ROUGE scores for the RLHF approach. We are fine-tuning the model to confirm the ground truth, which may seem overfit for samples and deviate from the generalization capability of the model. Instead of a single ground truth headline, if we generate at least three headlines for the given input text similar to RLHF, that may help us understand whether the model falls into the category of generating the most common headline among the three. The performance measures will also change with multiple headlines as the ground truth. We speculate that a human factor would be added if several headlines were used as the ground truth rather than a single headline that is more like a robotic approach.

6 Conclusion

We proposed a RAG-based fine-tuning of LLMs to generate headlines and numerical values through reasoning. The model was trained from end to end to optimize the output of results. The model was trained for 3 epochs for headline generation and 1 epoch for numerical reasoning. We would like to train the model for a longer number of epochs in the future to confirm whether the model can improve performance. The verification step used to validate the generated numbers by the model is very useful to improve the confidence of the generated headline. There may be several numbers in the news, but the extraction and verification of numbers that can contribute to headline generation is a more concentrated approach. The additional verification stage helped the human evaluator select our proposed methodology as the most efficient among the competitors. We would like to explore further the rationale steps followed by the model in COT and improve the model performance by taking advantage of mathematical operations. We

would like to divide operations into a subset of operations. A combination of results from multiple models is attempted without fully exploring the ensemble approach. We would like to explore the possibility of combining models at different levels such as input, architecture, and so on.

References

Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. 2019. Better rewards yield better summaries: Learning to summarise without references. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3110–3120, Hong Kong, China. Association for Computational Linguistics.

Pengshan Cai, Kaiqiang Song, Sangwoo Cho, Hongwei Wang, Xiaoyang Wang, Hong Yu, Fei Liu, and Dong Yu. 2023. Generating user-engaging news headlines. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3265–3280.

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Nquad: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2925–2929, New York, NY, USA. Association for Computing Machinery.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zijian Ding, Alison Smith-Renner, Wenjuan Zhang, Joel R Tetreault, and Alejandro Jaimes. 2023. Harnessing the power of llms: Evaluating human-ai text co-creation through the lens of news headline generation. *arXiv preprint arXiv:2310.10706*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

JiangLong He, Mamatha N, Shiv Vignesh, Deepak Kumar, and Akshay Uppal. 2022. Linear programming word problems formulation using ensemblecrf ner labeler and t5 text generator with data augmentations.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

LLAMA2. Llama2. `https://huggingface.co/docs/transformers/model_doc/llama2`. Accessed: 2024-02-14.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

MISTRAL. Mistral. `https://mistral.ai`. Accessed: 2024-02-14.

NumEval. Numeval competition. `https://sites.google.com/view/numeval/numeval`. Accessed: 2024-02-14.

OpenAI. Openai. `https://openai.com`. Accessed: 2024-02-14.

RAG. Rag. `https://huggingface.co/docs/transformers/en/model_doc/rag`. Accessed: 2024-02-19.

Rindranirina Ramamonjison, Timothy T. Yu, Raymond Li, Haley Li, Giuseppe Carenini, Bissan Ghaddar, Shiqi He, Mahdi Mostajabdaveh, Amin Banitalebi-Dehkordi, Zirui Zhou, and Yong Zhang. 2023. Nl4opt competition: Formulating optimization problems based on their natural language descriptions.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2020. Structure learning for headline generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9555–9562.

A Appendices

We present a few examples from the verification stage. The examples provide insight into the verification stage followed in the proposed method.

A.1 Examples without errors in the verification stage

A.1.1 Example 1

News: (Jan 23, 2008 3:50 PM) A blast in the northern Iraqi city of Mosul claimed at least 15 lives and left 132 wounded today, Reuters reports. The explosion came after Iraqi troops detonated a roadside bomb near a building used as a munitions dump by militants, a US general said. At least 15 tons of explosives are estimated to have been stored in the building. Witnesses reported the resulting explosion was one of the biggest ever seen in Mosul, which, along with other areas of Nineveh province, has become a center for Sunni extremist fighters following their expulsion from Baghdad and the western Anbar province. A separate attack near Kirkuk, also in the north, killed 16 today, Reuters adds.

Pred_seq: Iraqi Blast Kills 15 <NR> Copy(15) </NR> in Mosul

Processed_headline: Iraqi Blast Kills 15 in Mosul

The method searches for any XML tag in the generated sequence. Here, the verification has found a tag. The method searches for the number '15' in the text because it is copied directly from the news section. The number '15' is found by the verification stage and removes the XML tag present in the generated headline.

A.1.2 Example 2

News:(Apr 22, 2009 9:11 AM CDT) Sick of waiting for a grand jury to act, Kathleen Savio's family has filed a wrongful death suit against husband Drew Peterson, the Chicago Tribune reports. The grand jury, which expires next month, has been investigating the death of Savio, Peterson's third wife, and the disappearance of fourth wife Stacy Peterson since November 2007. The Savios have hired John Kelly, the high-powered attorney who won the Brown family's civil suit against OJ Simpson, to represent them. We felt that the current grand jury is going to bring forward an indictment, or they'll be dismissed and we can proceed, said Kelly, who acknowledged it was unusual to file a civil case while a grand jury is still meeting. Defense attorneys will now be able to subpoena evidence from the criminal case for the civil one—but the state action will take precedence if the grand jury does return an indictment.

Pred_seq: Savio Family Files Wrongful Death Suit Against Drew Peterson

Processed_headline: Savio Family Files Wrongful Death Suit Against Drew Peterson

The method searches for any XML tag in the generated sequence. Here, the verification passes the generated headline without any changes due to the absence of XML tags.

A.1.3 Example 3

News: (Jul 31, 2012 3:13 AM CDT) The American Red Cross is facing a blood donation shortage serious enough that some people may have to cancel

elective surgeries. Donations are at their lowest in 15 years. In June, the organization came up 50,000 units short. July isn't looking good, either—a particular problem given the increased number of accidents typical in the summer, when people travel, a rep tells NBC News. Storms in the eastern and midwestern US both increased demand and cut supply, as the Red Cross was forced to cancel drives. With students, who account for 20% of donations, donating far less in the summer, the problem is compounded. We normally try to keep a three-day supply on hand locally, and we are down to a one—day supply, warns an Ohio Red Cross worker. And the need never, ever goes away, notes another representative.

Pred_seq: Red Cross Faces Worst Blood Shortage in 15 <NR> Copy(15) </NR> Years

Processed_headline: Red Cross Faces Worst Blood Shortage in 15 Years

The method searches for any XML tag in the generated sequence. Here, the verification has found a tag. The method searches for the number '15' in the text because it is copied directly from the news section. The number '15' is found by the verification stage and removes the XML tag present in the generated headline.

A.2 Examples with errors in the verification stage

A.2.1 Example 1

News: (Dec 20, 2016 5:40 PM) Forty-three days after the election, all the votes have finally been tallied and certified. History will show Hillary Clinton beating Donald Trump by a final count of nearly 3 million votes, the Hill reports. According to a tweet Tuesday from the nonpartisan Cook Report, Clinton received 65,844,610 votes (48.2%) to Trump's 62,979,636 (46.1%). However with the Electoral College officially making Trump the 45th US president on Monday, history will also show Clinton as the second Democrat in the past five elections to win the popular vote but lose the presidency. Meanwhile, the Huffington Post reports Trump had the third worst popular-vote performance by a winning candidate on record.

Pred_seq: 43 <NR> Subtract(43,1) </NR> Days After Election, All Votes Have Been Counted

Issues: Wrong Subtracted Value ('43', '<NR> Subtract(43,1) </NR>') | (43, 42), Value 43 not found in Snippet

Selected_headline: History Will Show Clinton Won Popular Vote by 3M Votes

The method searches for any XML tag in the generated sequence. Here, the verification has found a tag. When we search for the number '43', it is not found in the news section. We discard the generated headline and select the headline generated by the BRIO model as the generated headline.

A.2.2 Example 2

News: (Jul 12, 2014 4:20 PM CDT) Here's a dream come true for couch potatoes: You're not going to have to stop watching this movie for an entire month when it is ultimately released. Ambiancé, by Swedish director Anders Weberg, will be a full 720-hour film to be screened only once, starting on New Year's Eve in the year 2020, on every continent simultaneously, and then destroyed, reports The Verge. The filmmaker, who says this movie will be his last, has just released a teaser on Vimeo that is a short 72 minutes long, or roughly the length of a normal film. The short trailer, due out in two years, will be 7 hours and 20 minutes long (notice a theme?), reports Smithsonian magazine. The final trailer, out in 2018, will be 72 hours long. The film, a surreal dream-like journey beyond places, according to the film's site, is a sort of memoir that gives an abstract nonlinear narrative summary of the artist's time spent with the moving image. And it's expected to set the record as the longest film ever made. The teaser is only available until July 20. (Meanwhile, it's not too late to catch part of the world's longest concert.)

Pred_seq: World's Longest Film Will Be Destroyed After 1 <NR> Subtract(2020,2014) </NR> Screening

Issues: Wrong Subtracted Value ('1', '<NR> Subtract(2020,2014) </NR>') |(1,6), Subtraction operation is failed

Selected_headline: Longest Film Ever Made, 720-Hour Movie to Be Made, Destroyed

The method searches for any XML tag in the generated sequence. Here, the verification has found a tag. But, the model has failed to complete the subtraction operation. We discard the generated headline and select the headline generated by the BRIO model as the generated headline.

A.2.3 Example 3

News: (Dec 20, 2008 3:27 PM) The US could double the number of its troops in Afghanistan to about 60,000 by next summer, the highest such estimate to date, Reuters reports. American troops

currently number 31,000, but joint chiefs chairman Mike Mullen said today that an additional 20,000 to 30,000 will be needed by the spring or early summer. We're going to fill that requirement so it's not a matter of if, but when, he said. The majority will be deployed to the fragile south of the country, and Mullen was candid about the dangers. When we get additional troops here, I think the violence level is going to go up, he said. The fight will be tougher. Mullen also stressed that normalizing relations between Pakistan and India would ease violence in Afghanistan, and that any military progress must go hand in hand with economic development.

Pred_seq: US Could Double Troops in Afghanistan by Summer '09 <NR> Subtract(2009,2008) </NR>

Issues: Wrong Subtracted Value ('09', '<NR> Subtract(2009,2008) </NR>') |(09,1), Value 2009 not found in Snippet

Selected_headline: US Could Send 60K More Troops to Afghanistan

The method searches for any XML tag in the generated sequence. Here, the verification has found a tag. When we search for the number '2009', it is not found in the news section. We discard the generated headline and select the headline generated by the BRIO model as the generated headline.

AlphaIntellect at SemEval-2024 Task 6: Detection of Hallucinations in Generated Text

Sohan Choudhury
KIIT, Bhubaneswar
sohan2004cc@gmail.com

Priyam Saha
Jadavpur University, Kolkata
priyam.saha2003@gmail.com

Subharthi Ray
Jadavpur University, Kolkata
subharthiray126@gmail.com

Shankha Shubhra Das
Jadavpur University, Kolkata
shankhasdas07@gmail.com

Dipankar Das
Jadavpur University, Kolkata
dipankar.dipnil2005@gmail.com

Abstract

One major issue in natural language generation (NLG) models is detecting hallucinations (semantically inaccurate outputs). This study investigates a hallucination detection system designed for three distinct NLG tasks: definition modeling, paraphrase generation, and machine translation. The system uses feedforward neural networks for classification and SentenceTransformer models for similarity scores and sentence embeddings. Even though the SemEval-2024 benchmark is showing good results, there is still room for improvement. Promising paths towards improving performance include considering multi-task learning methods, including strategies for handling out-of-domain data and minimizing bias, and investigating sophisticated architectures.

1 Introduction

AI hallucination refers to a phenomenon where a Large Language Model (LLM) - usually Generative AI or a computer vision tool produces nonsensical and inaccurate outputs (Maleki et al., 2024). Thus, this leads to fluent but inaccurate generations. The term 'hallucination' is usually associated with human or animal brains but from the standpoint of machines, hallucinations refer to these inaccurately produced outputs.

Hallucinations in generative models may arise due to multiple factors such as overfitting during model training, complexity of the model, and bias in training data. According to multiple surveys (Huang et al., 2023), Hallucinations in natural language models may arise primarily due to two reasons - Hallucinations due to data and hallucinations during modeling.

Hallucinations in AI models may prove to be a threat in multiple scenarios such as healthcare where a model may not be able to predict the existence of the exact condition that needs to be treated. Hallucinations in fluent over generations may also lead to the spread of misinformation.

The most pressing problem in the modern-day natural language generation landscape is that the existing metrics (Bandi et al., 2023) can mostly detect fluency in generation rather than accuracy.

To deal with this issue, several studies have explored different techniques, such as Knowledge Graph Integration, Bias Detection, and Mitigation (Rawte et al., 2023). Building upon this prior research, our work proposes training tailor-made deep learning models and using Transformer (Vaswani et al., 2017) based architectures to identify cases of hallucinations.

To deal with this scenario, a task (Mickus et al., 2024) was proposed to build a system to detect instances of hallucinations in generated text.

2 Task

The primary task was to build a hallucination detection system capable of detecting outputs that are grammatically sound but are semantically inaccurate concerning the provided source input - both with or without access to the model used to generate the outputs. This is essentially a binary classification task and we were provided with two tracks - model agnostic and model aware. Model agnostic refers to the track where one would have no access to the model used to generate the outputs and model aware refers to the track where the model was provided in the dataset.

3 Related Work

Recent advances in natural language processing (NLP) have resulted in the development of Transformer based models such as BERT and its specialized variations that have introduced efficiency and accuracy in several NLP tasks.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 952–958
June 20-21, 2024

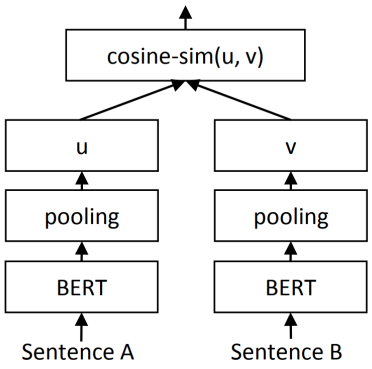


Figure 1: SBERT architecture to compute similarity scores

3.1 BERT

Bidirectional Encoder Representations from Transformers or BERT (Devlin et al., 2018), is an innovative machine learning technique for natural language processing (NLP). Researchers at Google AI Language created the adaptable BERT model in 2018, and it can perform more than 11 common NLP jobs, such as named entity recognition and sentiment analysis. Computers have never been very good at interpreting language. This requirement is attempted to be filled by NLP, a blend of languages, statistics, and machine learning. NLP activities required the usage of specialized models prior to BERT. With its cohesive approach and remarkable performance across a range of tasks, BERT transformed NLP.

3.2 SBERT

SBERT [1] or Sentence-Bert is a modified version of the BERT model which uses siamese [2] and triplet networks and is able to understand meaningful semantic embeddings in sentences. A common issue with BERT is, that the cross encoder setup of BERT takes up a lot of time and resources. To find the pair with the maximum similarity among n = 10,000 phrases, for example with BERT,

$$\frac{n(n-1)}{2} = 49{,}995{,}000$$

inference calculations are needed. This takes roughly sixty-five hours on a contemporary V100 GPU.

[1] http://arxiv.org/abs/1908.10084

[2] https://www.cs.cmu.edu/ rsalakhu/papers/oneshot1.pdf

4 Datasets

Since the task was divided into two tracks - model-aware and model-agnostic, we were provided with sets of two datasets. The model-aware dataset had a separate column for the generative model used to produce the outputs. The training dataset consisted of three natural language generation (NLG) tasks - Definition Modelling, Paraphrase Generation, and Machine Translation.

1. **Definition Modelling (DM)** - Clear and concise definition of concepts or terms generated by generative models.

2. **Paraphrase Generation (PG)** - Alternative wordings are generated that convey the same meaning as the input text.

3. **Machine Translation (MT)** - Translation of the text from one language to another while preserving fluency and meaning.

Each entry in the dataset comprises three text columns - hyp, src and tgt.

1. **hypothesis (hyp)** - Contains the generated text.

2. **source (src)** - The source text or the original text provided as input to the generative model for producing the hypothesis.

3. **target (tgt)** - Contains the correct generation output.

The trial dataset contains three columns dedicated to the labels. The 'labels' column contains the three most likely labels out of which the majority label is displayed in the 'label' column - which is either 'Hallucination' or 'Not Hallucination'. Finally, the 'p(Hallucination)' column comprises of probability values ranging from 0 to 1.

4.1 Trends in the dataset

The datasets provided include trial, validation (model-agnostic and model-aware), and test (model-agnostic and model-aware) sets, all represented by their respective figures (Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6).

The datasets contain almost an even distribution of DM and PG tasks. However, the number of rows with the task 'MT' is considerably less.

Certain entries in the dataset had no probability values and to avoid difficulties in the training process, we have dropped the rows.

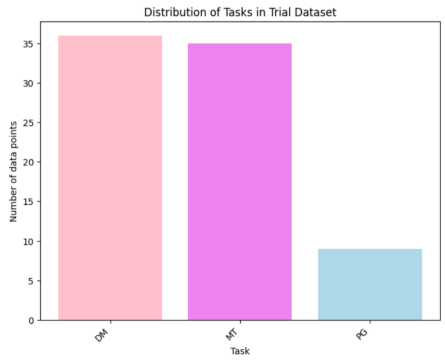


Figure 2: Trial Dataset

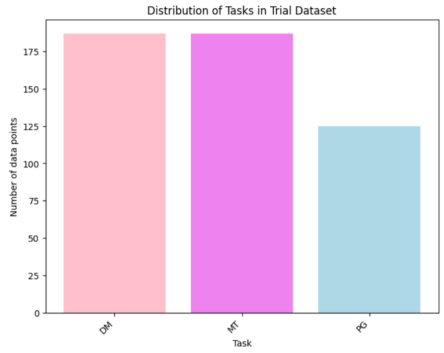


Figure 3: Validation Dataset (Model Agnostic)

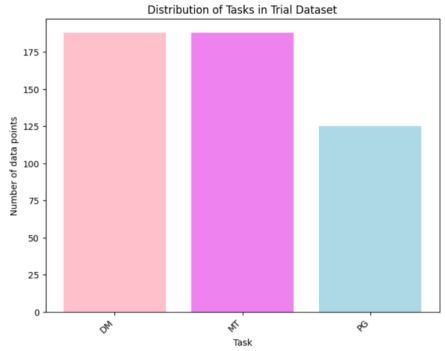


Figure 4: Validation Dataset (Model Aware)

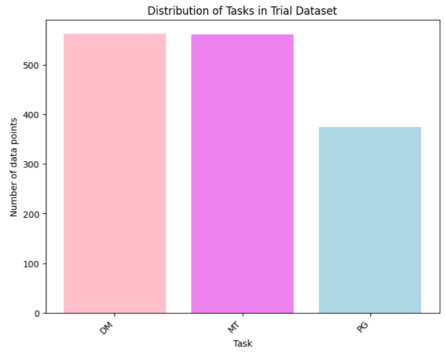


Figure 5: Test Dataset (Model Agnostic)

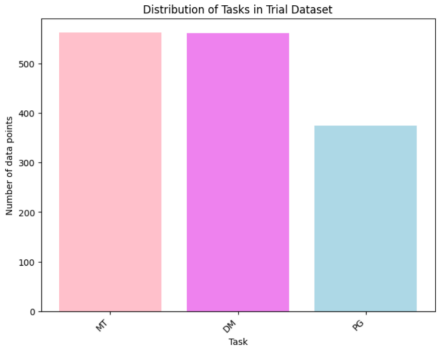


Figure 6: Test Dataset (Model Aware)

We determined that there was no viable utility for integrating the separate training dataset lacking annotations into our system. Furthermore, we were aware of the possible overfitting risk brought about by more unannotated data points. Consequently, we made the informed decision to refrain from its utilization.

5 Pre Processing

5.1 Data Cleaning

The text sections in the dataset have been processed by removing irrelevant elements in order to improve the efficiency of word embedding produced by the models used. The sentences under hyp, src, and tgt columns mainly contain prepositions and certain irrelevant expressions.

1. First the sentences are lowered by using Python's .lower() function.

2. In order to remove irrelevant expressions and prepositions, we have used Python's Regular Expression (re) library.

3. Finally, the sentences are stripped and split into individual words.

5.2 Labels

In order to make the training process more efficient, we converted the probability values in the 'p(Hallucination)' column into binary labels - if the probability value was more than or equal to 0.5, we converted it to 1 and otherwise it was labeled 0.

5.3 Tokenisation

We used 'all-mpnet-base-v2', a SentenceTransformer model to encode our sentences. The encoding process is comprised of three steps - tokenization, word embedding, and sentence embedding.

1. **Tokenization** - The sentence is split into individual tokens. This is done by the methods of stemming or lemmatization (Khyani and B S, 2021).

2. **Word Embedding** - Each token is then assigned a numerical vector. These are called word embeddings and they represent the semantic meaning of the word using information from a large text corpus.

3. **Sentence Embedding** - Finally the tokens are converted into a single vector representation for the entire sequence.

   Different models use different approaches to perform embedding.

   (a) **Mean pooling** - The average of all word embeddings is taken for the entire sentence. This method is useful in capturing the overall sentiment but may lead to a loss of semantic information.

   (b) **Weighted mean pooling** - Weights are assigned to the words using attention mechanisms to represent their importance in the sentence. This helps in prioritizing certain words and preserving semantic information.

   (c) **Transformers** - Transformer models consider the entire sentence and process the relationship between different words. Thus, contextualized embeddings capture the context more accurately.

6 Methodology

6.1 Experimental Setup

Since the generated texts have been divided into three tasks, we have divided the process into three branches.

For Definition modeling, we used 'all-mpnet-base-v2' [3] and for Paraphrase generation, we have used 'paraphrase-MiniLM-L6-v2' [4], both Sentence-Transformer models to encode the sentences and calculate two sets of cosine similarity values. One represents the similarity score between the hypothesis (hyp) and source (src) text and the other represents the similarity score between the source (src) and target (tgt) text. These similarity scores are converted into a numpy array using numpy's column stack for input into two sequential models respectively.

For Machine translation, we used 'all-MiniLM-L6-v2' [5], also a SentenceTransformers model to encode the sentences and produce embeddings. The Spearman correlation between the two sentences in the hypothesis and target columns is calculated. We chose these columns specifically as 'hyp' contains the English translation produced by the generative model and 'tgt' contains the correct translation. We used the SciPy library of Python for this metric. Finally, the correlation coefficients are pushed into a numpy array. In this case, we have used a different sequential model for training.

The input array is split into an 80:20 ratio for training and validation respectively and the test dataset was entirely used for producing the outputs. We have used a common pipeline for processing the entire dataset and then branched the input array according to the task label - if the task is 'DM' it was fed into the model prepared for definition modeling.

6.2 Model

We used three models for producing the outputs for the three tasks respectively.

6.2.1 Definition Modelling

Using the Tensorflow Keras framework, we created a deep-learning neural network. It has two densely concealed layers, each with 64 and 32 neurons. For the hidden layers, we employed ReLU activation functions, which give the model non-linearity (Kulathunga et al., 2021). This helps the model learn non-linear correlations and fortifies the neural network. To lessen overfitting, we have incorporated a dropout layer after each dense layer. Lastly, since this is a binary classification problem, we have utilized the sigmoid activation function for the output layer.

6.2.2 Paraphrase Generation

We designed a neural network architecture for the paraphrase generation type inputs comprising of an input layer accepting data with two features, followed by three hidden layers. The first layer consists of 128 neurons with ReLU activation, coupled with a dropout layer to mitigate overfitting. Subsequently, a 64-neuron layer employs ReLU

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[4] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2

[5] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Task Specific Model	Accuracy
Definition Modelling	0.67
Paraphrase Generation	0.74
Machine Translation	0.83

Table 1: Evaluation of Individual Models

activation, batch normalization, and dropout regularization. Similarly, the third hidden layer integrates 32 neurons, ReLU activation, batch normalization, and dropout regularization. The outputs from the second and third layers are concatenated before feeding into a single-neuron output layer with sigmoid activation, typical for binary classification. This architecture is optimized using the Adam optimizer with a learning rate of 1e-4 and binary cross-entropy loss, while early stopping is applied during training to prevent overfitting. The model's configuration demonstrates a structured approach to feature extraction and classification, tailored for paraphrase generation types input data.

6.2.3 Machine Translation

The model for machine translation harnesses the formidable capabilities of the all-MiniLM-L6-v2 Sentence Transformer, designed to adeptly encode semantic nuances within input sentences into dense embeddings. These embeddings undergo meticulous examination through Spearman correlation (halo), discerning their intrinsic similarities. Post-normalization, they serve as inputs to a meticulously designed neural network architecture, capitalizing on ReLU activation functions for intricate feature extraction. Culminating in a sigmoid activation layer, the network adeptly estimates the probability of sentence hallucinations, embodying a rigorously scientific approach to classification.

6.2.4 Evaluation of Task Specific Models

This section presents an evaluation of three task-specific models trained for Definition Modelling (DM), Paraphrase Generation (PG), and Machine Translation (MT) tasks, respectively. Each model is assessed based on its training accuracy, providing insights into its performance on the training data.

6.3 Loss

We use binary cross-entropy (BCE) (Ruby and Yendapalli, 2020) as our loss function as it measures the difference between the predicted probability and the true binary label (0 or 1). BCE is not affected by class imbalance, which occurs when one class has noticeably fewer samples than the other, in contrast to Mean Squared Error (MSE). This guarantees that the model concentrates on efficiently learning both classes.

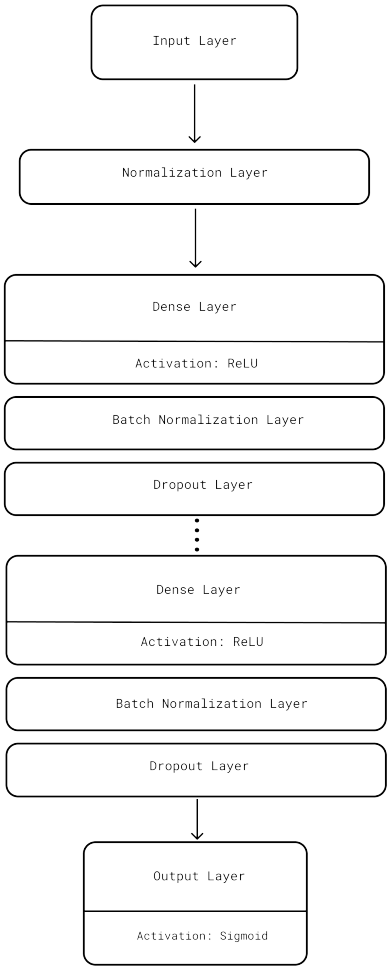


Figure 7: Generalised Architecture of the Model

6.4 Optimizer

We use the Adam (Kingma and Ba, 2017) optimizer for our neural networks as Adam is effective at traversing the loss landscape because it combines momentum and adjustable learning rates. It strikes a balance between exploration and exploitation, enabling the model to iteratively identify areas of high performance and improve its solutions.

A generalized representation of the model is visualized in Figure 7.

7 Evaluation

The predicted probability values generated by the model were translated into binary labels using a

Track	Accuracy	Rho
Model-Agnostic	0.654	0.294608108
Model-Aware	0.7113333333	0.4264291384

Table 2: Evaluation results

threshold approach. Data points with a predicted probability of 0.5 or higher were assigned the label "Hallucination," while those below 0.5 were labeled "Not Hallucination." These labels were then saved in a JSON file conforming to the specified format.

The SemEval-2024 task had two measures to evaluate the performance:

1. the accuracy that the system reached on the binary classification.

2. the Spearman correlation of the systems' output probabilities with the proportion of the annotators marking the item as overgenerating.

   It is given by:

$$\rho_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}$$

   where d_i is the difference in ranks for item i and n is the total number of items.

Also, the submissions were divided into model agnostic and model aware tracks. Our model placed 31st out of all entries in the model-aware track and 41st out of all entries in the model-agnostic track.

8 Conclusion and Future Work

This paper explored detecting hallucinations in natural language generation (NLG) outputs using specialized models for definition modeling, paraphrase generation, and machine translation tasks - both while having access to the models used to generate the sentences and without. We used transformer-based models for calculating the similarity scores as they outperform other models such as Universal Sentence Encoder (USE) (Cer et al., 2018) and Doc2Vec (Lau and Baldwin, 2016).

While there have been previous studies on hallucination detection, our approach offers several key novelties that have contributed to its effectiveness.

- **Task-specific Models**: Instead of the one-size-fits-all approach, we built specific models for each task to better capture their unique characteristics. This customization aids in efficiently extracting features crucial for identifying hallucinations.

- **Transformer-based Similarity Scores**: To compute sentence similarity scores, we made use of SentenceTransformer, a Transformer based model. These models do better at capturing contextual information and fine-grained semantic relationships inside phrases than other models.

Several avenues exist for further development of our hallucination detection system. To enhance the performance of our model, we advise investigating data augmentation techniques, as transformer-based models have a large thirst for data. To increase the model's robustness and durability, we also suggest using adversarial training and exploring more advanced deep learning architectures.

This project has been possible due to the contributions of Sohan Choudhury, who developed the architecture for the definition modelling task, Priyam Saha, who created the paraphrase generation model, and Subharthi Ray, who built the machine translation model. We are also deeply grateful for the insightful guidance and mentorship provided by Shankha Shubhra Das and Dr. Dipankar Das throughout this journey.

References

Ajay Bandi, Pydi Venkata Satya Ramesh Adapa, and Yudu Eswar Vinay Pratap Kumar Kuchi. 2023. The power of generative ai: A review of requirements, models, inputndash;output formats, evaluation metrics, and challenges. *Future Internet*, 15(8).

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Jacob Devlin, Kenton Lee Ming-Wei Chang, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting

Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Divya Khyani and Siddhartha B S. 2021. An interpretation of lemmatization and stemming in natural language processing. *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology*, 22:350–357.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Nalinda Kulathunga, Nishath Rajiv Ranasinghe, Daniel Vrinceanu, Zackary Kinsman, Lei Huang, and Yunjiao Wang. 2021. Effects of nonlinearity and network architecture on the performance of supervised neural networks. *Algorithms*, 14(2).

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. *CoRR*, abs/1607.05368.

Negar Maleki, Balaji Padmanabhan, and Kaushik Dutta. 2024. Ai hallucinations: A misnomer worth clarifying.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *Preprint*, arXiv:2309.05922.

Usha Ruby and Vamsidhar Yendapalli. 2020. Binary cross entropy with deep learning technique for image classification. *International Journal of Advanced Trends in Computer Science and Engineering*, 9.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

YSP at SemEval-2024 Task 1: Enhancing Sentence Relatedness Assessment using Siamese Networks

Yasamin Aali
Alzahra University
yasamin.aali01@gmail.com

Sardar Hamidian
GWU
sardar@gwu.edu

Parsa Farinneya
University of Toronto
parsa.farinneya
@mail.utoronto.ca

Abstract

In this paper we present the system for Track A in the SemEval-2024 Task 1: Semantic Textual Relatedness for African and Asian Languages (STR). The proposed system integrates a Siamese Network architecture with pre-trained language models, including BERT, RoBERTa, and the Universal Sentence Encoder (USE). Through rigorous experimentation and analysis, we evaluate the performance of these models across multiple languages. Our findings reveal that the Universal Sentence Encoder excels in capturing semantic similarities, outperforming BERT and RoBERTa in most scenarios. Particularly notable is the USE's exceptional performance in English and Marathi. These results emphasize the importance of selecting appropriate pre-trained models based on linguistic considerations and task requirements.

1 Introduction

Semantic relatedness is a fundamental measure in natural language processing, providing a detailed assessment of how closely related two pieces of text are on a semantic level. This metric has wide-ranging significance in NLP tasks such as information retrieval, question answering, and text summarizing, contributing to the understanding of textual connections and improving algorithmic performance.

The importance of determining meaning through quantifying relatedness has been acknowledged within linguistic discussions for many years. Automating the determination of connected meanings holds significant value across diverse applications like evaluating sentence representation methods or supporting question-answering systems as well as summarizing processes.

However, the application of measuring relatedness to non-English languages presents substantial challenges due to disparities in linguistic resources and annotated datasets. In contrast to English many languages lack comprehensive lexical or syntactic resources which creates challenges in inaccurately capturing semantic subtleties and establishing cross-language associations. Additionally, variations across different languages such as morphological, syntactic, and semantic also add complexity to developing language-independent models for expressing relatedness Overcoming these obstacles requires extensive research and effective resource development tailored to different linguistic contexts ensuring that measures offered by the semantic similarity can be effectively applied across different languages. Here are some examples of the score of semantic relatedness of two sentences in three languages:

English: *"You figure this out all by yourself, did you?*
did you find all this on your own?" Score: 0.88

Spanish: *"Jean Hebb Swank es una astrofísica conocida por sus estudios sobre agujeros negros y estrellas de neutrones.*
Bajo la supervisión de Steve Frautschi, obtuvo su doctorado en física en 1967." Score: 0.52

Kinyarwanda: *"East Africa's Got Talent ku nshuro yayo ya mbere u Rwanda ni kimwe mu bihugu byemerewe kuyitabira aho rwahuriye n'ibindi birimo Uganda, Tanzania na Kenya.*
Iya mbere ni umubano mwiza uri hagati ya Mali n'u Rwanda, ndetse u Rwanda ni kimwe mu bihugu bifite abapolisi bagiye kugarura amahoro muri Mali." Score: 0.09

The rest of this paper is structured as follows: Section 2 introduces the problem statement and provides a summary of related works. Sections 3 and 4 detail the system description and experimental setup, respectively. The evaluation results are outlined in Section 5, followed by our conclusion

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 959–963

Language	Train	Dev	Test
English	0.75	0.79	**0.82**
Amharic	0.63	0.57	**0.64**
Algerian Arabic	0.44	**0.53**	0.4
Spanish	0.65	**0.66**	0.64
Hausa	**0.39**	0.38	**0.39**
Kinyarwanda	0.27	0.1	**0.31**
Marathi	0.58	0.65	**0.69**
Telugu	0.61	**0.75**	0.64

Table 1: The table provides a summary of the model's performance across different languages, for train, dev and test set, highlighting its strengths and weaknesses. It mentions the highest scores achieved in English, as well as the performance in Marathi and Telugu compared to English.

in Section 6.

2 Background

2.1 SemEval Task Description

We perform our experiments on data from the first subtask (supervised) of task 1 of SemEval-2024 (Ousidhoum et al., 2024b). We used 5,500 samples with 8 language pairs in the official training set (Ousidhoum et al., 2024a). The goal of the task is to predict the semantic textual relatedness between sentence pairs in different languages. The similarity score of pairs of articles in the provided dataset ranged from 0 to 1, with higher scores indicating higher semantic relatedness. To address the challenges of measuring semantic relatedness in non-English languages, research efforts need to focus on expanding resources and developing language-specific models.

2.2 Related work

Previous approaches to semantic relatedness have been categorized into knowledge-based and corpus-based methods. Knowledge-based methods use lexical resources like WordNet (Miller, 1995) to measure definitional overlap (Lesk, 1986), term distance within taxonomies, and term depth as specificity measures, among others. Knowledge-based methods are widely used in NLP applications, including word sense disambiguation and automatic summarization. The sources of knowledge utilized in these methods encompass various elements such as fuzzy logic, domain knowledge, Knowledge Graphs, ontologies, The Wikipedia among others. WordNet is an English language lexicon that arranges ideas into a conceptual structure. Its purpose is to represent the meaning of English words by categorizing synonyms and various relationships, both taxonomic and non-taxonomic. While semantic similarity quantifies specific likeness, relatedness provides a broader measure that encompasses connectedness as well.

On the other hand, corpus-based measures utilize probabilistic approaches such as Latent Semantic Analysis (Landauer et al., 1997), Explicit Semantic Analysis (Gabrilovich et al., 2007) and Salient Semantic Analysis (Hassan and Mihalcea, 2009), to decode word semantics based on contextual information observed in raw text. Corpus-based methods involve statistically analyzing large text corpora to quantify semantic similarities employing distributional semantics principles that capture contextual information within the text itself. Among these approaches is Latent Semantic Analysis, which uses a singular value decomposition technique to minimize word co-occurrence pattern matrix dimensionality within a corpus successfully applied among various natural language processing tasks including text classification and information retrieval. Another notable method utilizes random projection for mapping words onto high-dimensional spaces with cosine similarity vectors; termed Random Indexing, it outperforms LSA in particular tasks showing utility across diverse NLP applications like word sense disambiguation. In addition, researchers are exploring techniques leveraging the web as a corpus leading toward a branch known as web intelligence.

Neural networks have gained increasing significance in evaluating semantic relatedness due to their ability to comprehend intricate connections and subtleties in meaning from extensive data. Traditional approaches for assessing semantic relatedness often depend on manually crafted features or knowledge-based methods, which may have limitations in capturing the complete spectrum of semantic relationships between words and sentences. In contrast, neural networks can undergo training using large datasets to acquire contextualized representations of words and sentences. These representations capture the subtle meanings typically overlooked by traditional methods, enabling neural networks to achieve cutting-edge performance on tasks related to semantic relatedness. Furthermore, neural networks can be fine-tuned for specific purposes, enhancing their precision and efficiency. Furthermore, recent advancements in deep learning and

neural network technology have demonstrated encouraging outcomes when it comes to measuring semantic relatedness. For example, several pretrained language models like BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) have been tailored for semantic relatedness applications with state-of-the-art performance achieved on standard benchmark datasets.

Semantic relatedness in neural networks has been greatly impacted by the emergence of transformers, leading to a significant transformation in natural language processing (Gagliardi and Artese, 2023). Unlike traditional methods relying on manual features and statistical models, transformers utilize self-attention to dynamically allocate attention across input elements, effectively capturing long-term dependencies within language data. Current approaches for semantic relatedness mainly involve using powerful models like transformers to encode sentences into embeddings and then computing their similarity score using metrics such as cosine similarity. These advancements in deep learning and neural network technology have enabled the development of powerful models that can accurately measure semantic similarity and relatedness between sentences, surpassing the capabilities of traditional methods.

3 System Overview

The measurement of semantic relatedness involves assessing the connection or correlation between words or phrases, regardless of their meanings. Recently, deep learning models like convolutional neural networks and recurrent neural networks have garnered attention in measuring semantic relatedness. These models excel at capturing intricate patterns and interdependencies in textual data, thereby enhancing performance across a range of natural language processing tasks. For instance, Siamese networks make use of CNN or RNN structures to compare embeddings and gauge the relatedness between sentences or short texts. Attention mechanisms have also been integrated into these models to improve focus on crucial semantic elements (Sharma, 2023).

Siamese architectures are effective because they use the same model to handle similar inputs, and makes it easier to compare sentence pairs and reduces the number of parameters that need training, requiring less data and making them less susceptible to overfitting (Ranasinghe et al., 2019).

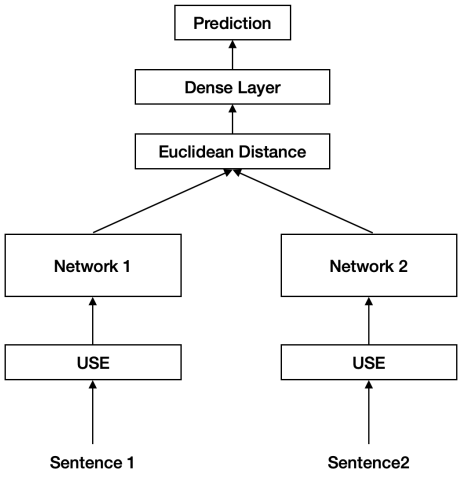


Figure 1: Basic structure of our system.

We implement a comprehensive system for evaluating sentence relatedness using a Siamese Network architecture coupled with Universal Sentence Encoder (USE) embeddings (Cer et al., 2018). The system comprises several distinct components, each contributing to the overall functionality and effectiveness of the model.

The Universal Sentence Encoder model is loaded from TensorFlow Hub [1], enabling the generation of high-quality embeddings for the input sentences. These embeddings capture semantic information, thereby facilitating the comparison and analysis of sentence similarity. Both the training and validation sentences are encoded into USE embeddings, which are then converted to NumPy arrays for seamless integration with the Siamese Network architecture.

The core of the system lies in the construction and training of the Siamese Network model (Figure 1). Utilizing TensorFlow's functional API, the model is designed to accept two input embeddings corresponding to pairs of sentences. It computes the Euclidean distance between these embeddings and passes the result through a Dense layer with sigmoid activation to predict the similarity score between the sentences. By employing Mean Squared Error (MSE) loss and the Adam optimizer, the model is trained on the training data, with performance monitored using the validation set.

Our code is available on GitHub [2]

[1] https://tfhub.dev/google/universal-sentence-encoder/4

[2] https://github.com/yasaminaali/Enhancing-Sentence-Relatedness-Assessment-using-Siamese-Networks

Models	English	Amharic	Alg Arabic	Spanish	Hausa	Kinyarwanda	Marathi	Telugu
USE	**0.75**	0.63	0.44	0.65	0.39	0.27	0.58	0.61
Bert	0.42	0.1	0.4	**0.5**	0.04	0.01	0.32	0.21
Roberta	0.38	0.2	0.31	0.46	0.01	0.1	**0.51**	0.24

Table 2: This table summarizes the key findings of the evaluation, highlighting the varied performances of different models across multiple languages. It specifically mentions the strengths of BERT in English and Spanish, RoBERTa's excellence in Marathi and Spanish, and the consistently exceptional results of the Universal Sentence Encoder across most languages in the training set.

4 Experimental Setup

4.1 Data Split

For all of our experiments, we split the task's training set using an 80/20 train/dev split, and we used the official development set as a test set.

4.2 Pre-processing

Pre-processing improves data quality, eliminates irrelevant information, and makes data more suitable for the calculation of the semantic relatedness score. This entails removing punctuation, numbers, and special characters such as # and $. Contractions are expanded to their full forms, and all text is converted to lowercase for consistency in processing.

4.3 Evaluation Metrics

The evaluation metric for task 1 is the Spearman Correlation between the predicted similarity scores and the human-annotated gold scores, with a range from 0 to 1 (from least to most correlated), which helps to determine how well the predicted scores align with human judgments.

5 Results

Our rankings show that in certain languages, our method closely matches the baseline performance. For example, in English, our rank is 0.82 compared to the baseline of 0.83, and in Spanish, our rank stands at 0.64 compared to the baseline of 0.7. Therefore, our method demonstrates significant effectiveness across different languages.

The evaluation of models across different languages showed varied performances. BERT demonstrated strong performance in English and Spanish, while RoBERTa excelled in Marathi and Spanish. However, the Universal Sentence Encoder consistently delivered exceptional results across most languages in the training set (Table 2).

Specifically, the USE model achieved the highest scores in English with 0.75 on the training set and 0.82 on the test set, indicating remarkable performance. The top overall score reached 0.86, showcasing its effectiveness in this task as well. Following English, Marathi exhibited the second-best performance at a score of 0.69 (Table 1).

5.1 Error Analysis

The model struggled with Kinyarwanda, Hausa and Algerian Arabic, hinting at relatively poor performance for this languages (Table 1).

To gain a deeper understanding of our model's performance, we compare its predictions with the test labels in English. Upon observation, it is evident that the model struggles when calculating the lowest semantic relatedness between sentences.

6 Conclusion

The system presented offers a strong framework for evaluating sentence similarity by integrating a Siamese Network architecture with Universal Sentence Encoder embeddings. A comprehensive overview of the system's components and processes, including data preprocessing, model construction, and training, demonstrates that the system effectively utilizes advanced techniques in natural language processing to make accurate similarity predictions. The evaluation results reveal the superior performance of the Universal Sentence Encoder across multiple languages, outperforming pre-trained models like BERT and RoBERTa in most scenarios. Notably, the system excelled in English and Marathi, demonstrating its versatility and effectiveness across diverse linguistic contexts. Further optimization and refinement of the system may enhance its performance in under-performing languages as well as broaden its applicability in real-world scenarios, ultimately advancing the field of natural language processing and facilitating a wide range of practical applications.

References

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Evgeniy Gabrilovich, Shaul Markovitch, et al. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611.

Isabella Gagliardi and Maria Teresa Artese. 2023. Ensemble-based short text similarity: An easy approach for multilingual datasets using transformers and wordnet in real-world scenarios. *Big Data and Cognitive Computing*, 7(4):158.

Samer Hassan and Rada Mihalcea. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 1192–1201.

Thomas K Landauer, Darrell Laham, Bob Rehder, and Missy E Schreiner. 1997. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *Proceedings of the 19th annual meeting of the Cognitive Science Society*, pages 412–417.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *Preprint*, arXiv:2402.08638.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orăsan, and Ruslan Mitkov. 2019. Semantic textual similarity with siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1004–1011.

Kabir Sharma. 2023. 30 years of research on semantic similarity measurement.

NootNoot At SemEval-2024 Task 6: Hallucinations and Related Observable Overgeneration Mistakes Detection

Sankalp Bahad[1]
IIIT Hyderabad
sankalp.bahad@research.iiit.ac.in

Yash Bhaskar[1]
IIIT Hyderabad
yash.bhaskar@research.iiit.ac.in

Parameswari Krishnamurthy[2]
IIIT Hyderabad
param.krishna@iiit.ac.in

Abstract

Semantic hallucinations in neural language generation systems pose a significant challenge to the reliability and accuracy of natural language processing applications. Current neural models often produce fluent but incorrect outputs, undermining the usefulness of generated text. In this study, we address the task of detecting semantic hallucinations through the SHROOM (Semantic Hallucinations Real Or Mistakes) dataset, encompassing data from diverse NLG tasks such as definition modeling, machine translation, and paraphrase generation. We investigate three methodologies: fine-tuning on labelled training data, fine-tuning on labelled validation data, and a zero-shot approach using the Mixtral 8x7b instruct model. Our results demonstrate the effectiveness of these methodologies in identifying semantic hallucinations, with the zero-shot approach showing competitive performance without additional training. Our findings highlight the importance of robust detection mechanisms for ensuring the accuracy and reliability of neural language generation systems.

1 Introduction

The modern NLG landscape is plagued by two interlinked problems: On the one hand, our current neural models have a propensity to produce inaccurate but fluent outputs; on the other hand, our metrics are most apt at describing fluency, rather than correctness. This leads neural networks to "hallucinate", i.e., produce fluent but incorrect outputs that we currently struggle to detect automatically. For many NLG applications, the correctness of an output is however mission critical. For instance, producing a plausible-sounding translation that is inconsistent with the source text puts in jeopardy the usefulness of a machine translation pipeline. With our shared task, we hope to foster the growing interest in this topic in the community.

With SHROOM (Mickus et al., 2024) we adopt a post hoc setting, where models have already been trained and outputs already produced: participants will be asked to perform binary classification to identify cases of fluent overgeneration hallucinations in two different setups: model-aware and model-agnostic tracks. That is, participants must detect grammatically sound outputs which contain incorrect or unsupported semantic information, inconsistent with the source input, with or without having access to the model that produced the output. To that end, we will provide participants with a collection of checkpoints, inputs, references and outputs of systems covering three different NLG tasks: definition modeling (DM), machine translation (MT) and paraphrase generation (PG), trained with varying degrees of accuracy. The development set will provide binary annotations from at least five different annotators and a majority vote gold label.

2 Dataset

SHROOM (Semantic Hallucinations Real Or Mistakes) dataset, aimed at addressing the challenge of detecting semantic hallucinations in neural language generation systems. SHROOM encompasses data from three diverse NLG tasks: data modeling (DM), machine translation (MT), and paraphrase generation (PG). Each task presents unique nuances in identifying semantic inaccuracies within generated text.

2.1 Model Aware and Agnoustic Data

There were two types of data entries, one in which the model information was present, which was the model aware dataset and the other where the model information was absent, the model agnousitc dataset.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 964–968

Example 1: Definition Modeling (DM)
Source: The sides of the casket were covered with heavy black broadcloth, with velvet caps, presenting a deep contrast to the rich surmountings. What is the meaning of surmounting? **Target:** A decorative feature that sits on top of something. **Hypothesis:** A sloping top. **Model:** ltg/flan-t5-definition-en-base **Label:** Hallucination **Probability (Hallucination):** 0.6

Example 2: Definition Modeling (DM)
Source: And the mower whets his sithe, What is the meaning of whet? **Target:** To hone or rub on with some substance, as a piece of stone, for the purpose of sharpening – see whetstone. **Hypothesis:** To cause to whirl. **Model:** ltg/flan-t5-definition-en-base **Label:** Hallucination **Probability (Hallucination):** 1.0

Example 3: Definition Modeling (DM)
Source: The corporation was accused of unethical behavior for knowingly producing a product suspected of harming health. What is the meaning of unethical? **Target:** Not morally approvable; morally bad; not ethical. **Hypothesis:** Not ethical; not conforming to ethical principles. **Model:** ltg/flan-t5-definition-en-base **Label:** Not Hallucination **Probability (Hallucination):** 0.0

Example 4: Paraphrase Generation (PG)
Source: We must create a society where everyone is able to enjoy a good education. **Hypothesis:** Everyone is capable of enjoying a good education in a society. **Model:** tuner007/pegasus$_{p}araphrase$ **Label:** Hallucination **Probability (Hallucination):** 0.8

Example 5: Paraphrase Generation (PG)
Source: Schooling is a fundamental issue that we should today reaffirm. **Hypothesis:** We should reiterate the importance of schooling. **Model:** tuner007/pegasus$_{p}araphrase$ **Label:** Not Hallucination **Probability (Hallucination):** 0.2

Example 6: Machine Translation (MT)
Source: Malo osungilako asilikali ankhondo amaluso osiyanasiyana ku depatimenti ya zachitetezo yaku U.S. ikutsata ziduswa. **Target:** The United States Strategic Command of the U.S. Department of Defense office is tracking the debris. **Hypothesis:** The U.S. Department of Defense's military intelligence facility is tracking the targets. **Model:** facebook/nllb-200-distilled-600M **Label:** Hallucination **Probability (Hallucination):** 1.0

Table 1: Examples from SHROOM Val dataset

2.2 Data Analysis

The dataset compilation involved sourcing data from a variety of sources to ensure its robustness and generalizability. For DM, definitions were gathered from various domains, covering a wide range of topics. MT data consisted of parallel corpora from multiple language pairs to capture translation nuances effectively. Finally, for PG, a collection of sentences and corresponding paraphrases from various genres was curated to represent natural language variation comprehensively.

2.3 Annotation

Annotating the dataset for semantic hallucinations followed a binary scheme, where each instance was labeled by 5 annotators as either containing semantic hallucinations or being free of such errors. To ensure the reliability of annotations, each instance underwent assessment by at least five annotators, with a majority vote determining the gold label.

2.4 Dataset Statistics

The SHROOM dataset comprises of multiple instances across all tasks. The distribution of instances for each NLG task is summarized below:

NLG Task	Train Set	Test Set
Definition Modeling (DM)	10000	563
Machine Translation (MT)	10000	562
Paraphrase Generation (PG)	10000	375

Table 2: Distribution of Instances by Task

For the Model Agnostic Dataset and Model Aware Dataset, each has:

Validation Set (Labelled):

NLG Task	Instances
Data Modeling	187
Paraphrase	125
Machine Translation	187

Train Set (Unlabelled):

NLG Task	Instances
Data Modeling	10000
Paraphrase	10000
Machine Translation	10000

Test Set:

NLG Task	Instances
Data Modeling	563
Paraphrase	375
Machine Translation	562

2.5 Example Instances

Table 2 provides examples from the SHROOM dataset, showcasing instances with and without semantic hallucinations for each NLG task.

Our participation in this shared task involves leveraging the SHROOM dataset to develop and evaluate models for detecting semantic hallucinations in NLG systems. This dataset serves as a valuable resource for benchmarking and advancing research in this area.

3 Methodology

Our Methodology involved first Labelling the Training Data, fine tune a model on the test data then evaluating the model on test data. We chose Roberta-base as our base model for fine tuning as XLM-RoBERTa is a multilingual language model optimized for classification tasks. It is pretrained on massive multilingual data, and has a robust architecture and performance enable efficient fine-tuning across diverse text classification problems with state-of-the-art accuracy. For Labelling the Training Data, we used Mixtral 8x7B (Jiang et al., 2024), specifically mixtral-8x7b-instruct-v0.1.Q5_K_M.gguf

3.1 Labelling Training Dataset

The prompt (Zamfirescu-Pereira et al., 2023) used for Labelling Training Dataset using Mixtral-8x7b-instruct is:

```
if task == "PG":
    context = f"Context: {src}"
else: # i.e. task == "MT" or task == "DM":
    context = f"Context: {tgt}"

sentence = f"Sentence: {hyp}"
message = f"{context}\n{sentence}\nIs
the Sentence supported by the Context
above?
Answer using ONLY yes or no:"
```

```
prompt = f"[INST] {message} [/INST]"
```

3.2 Finetune Roberta-base on the Mixtral labelled train dataset

We chose to fine-tune Roberta-base on the Mixtral labeled train dataset to adapt the model specifically for the task of detecting semantic hallucinations. The Mixtral labeled training dataset provided binary labels indicating whether a given sentence exhibited semantic hallucinations or not. The probability label for hallucination ranged from 0 to 1, derived from the log probability of the Mixtral model output. Therefore, we formulated the task as a binary classification problem: distinguishing between sentences containing semantic hallucinations and those that do not.

During fine-tuning, we modified the last layer of Roberta-base to accommodate the binary classification task. We used techniques such as cross-entropy loss and gradient descent to update the model's parameters based on the labeled training data. By fine-tuning on the Mixtral labeled dataset, we aimed to enhance Roberta-base's ability to identify semantic hallucinations in natural language generation outputs.

3.3 Finetune Roberta-base on the Pre-Annotated Data

In addition to fine-tuning on the Mixtral labeled train dataset, we performed fine-tuning on preannotated data, specifically the development dataset. This dataset had been annotated by five annotators, and each instance was assigned a probability label for hallucination ranging from 0 to 1 in increments of 0.2. The probability labels were based on the consensus among the annotators.

To leverage the fine-grained annotations provided by multiple annotators, we formulated the task as a multi-class classification problem. We fine-tuned Roberta-base (Conneau et al., 2020) to classify instances into one of six categories corresponding to the six probability levels (0, 0.2, 0.4, 0.6, 0.8, or 1). This approach allowed the model to learn from the nuanced annotations provided by the annotators and make more nuanced predictions about the presence of semantic hallucinations.

By fine-tuning Roberta-base on both the Mixtral labeled train dataset and the preannotated development dataset, we aimed to create a robust model capable of accurately detecting semantic hallucinations across a range of natural language generation tasks and datasets.

4 Results

We present the results of our experiments using three different methodologies for detecting semantic hallucinations in neural language generation systems.

4.1 Methodology 1: Fine-tune on the labelled Training Data (2 Class)

We fine-tuned our model on the labelled training data, treating the task as a binary classification problem. The results over multiple epochs are summarized in Table 3. We observed an improvement in both agnostic and aware accuracy over epochs, with agnostic accuracy reaching 76.47% and aware accuracy reaching 61.27% by the third epoch. However, the Matthews correlation coefficient (rho) showed less consistent improvement, with agnostic rho peaking at 0.58 and aware rho at 0.38 in the second epoch.

Epoch	**Agnostic Acc.**	**Aware Acc.**
1	0.753	0.609
2	0.759	0.601
3	0.765	0.613
Epoch	**Agnostic ρ**	**Aware ρ**
1	0.568	0.346
2	0.580	0.381
3	0.584	0.355

Table 3: Results for Methodology 1: Fine-tune on labelled Training Data (2-Class)

4.2 Methodology 2: Fine-tune on the labelled Validation Data (6 Class)

In this methodology, we fine-tuned the model on the labelled validation data, treating the task as a six-class classification problem. Results are presented in Table 4. Agnostic accuracy fluctuated around 45-51% over different epochs, while aware accuracy showed similar fluctuations around 47-58%. Matthews correlation coefficient (rho) varied between 0.43 and 0.52 for agnostic classification and between 0.48 and 0.52 for aware classification.

4.3 Methodology 3: Zero-shot Mixtral 8x7b

For the zero-shot approach (Yue et al., 2023), where we directly applied the Mixtral 8x7b model without fine-tuning, results are shown in Table 5. Agnostic accuracy achieved 78.73%, while aware accuracy

Epoch	Agnostic Acc.	Aware Acc.
3	0.515	0.578
5	0.473	0.487
10	0.449	0.483
15	0.463	0.473
Epoch	**Agnostic ρ**	**Aware ρ**
3	0.477	0.490
5	0.477	0.490
10	0.502	0.524
15	0.434	0.512

Table 4: Results for Methodology 2: Fine-tune on labelled Validation Data (6-Class)

reached 77.73%. The Matthews correlation coefficient (rho) for agnostic classification was 0.50, and for aware classification, it was 0.48.

Overall, the zero-shot approach demonstrated competitive performance compared to fine-tuning on labelled data, indicating the effectiveness of the Mixtral 8x7b model in detecting semantic hallucinations without additional training.

Approach	Agnostic Acc.	Aware Acc.
Zero-shot	0.787	0.777
Approach	**Agnostic ρ**	**Aware ρ**
Zero-shot	0.499	0.485

Table 5: Results for Methodology 3: Zero-shot Mixtral 8x7b

5 Conclusion

In this study, we investigated three different methodologies for detecting semantic hallucinations in neural language generation systems. We fine-tuned a model using labelled training data, labelled validation data, and also explored a zero-shot approach using the Mixtral 8x7b instruct model.

Our results indicate that fine-tuning on labelled data, whether it is the training data or the validation data, led to improvements in both agnostic and aware accuracy over multiple epochs. However, the effectiveness of fine-tuning on validation data seemed to diminish as the number of epochs increased, suggesting potential overfitting.

Interestingly, the zero-shot approach using the Mixtral 8x7b instruct model achieved competitive performance compared to fine-tuning on labelled data. This indicates the robustness of the Mixtral model in detecting semantic hallucinations without additional training.

Overall, our findings suggest that while fine-tuning on labelled data can lead to improvements in detection accuracy, the zero-shot approach with pre-trained models like Mixtral 8x7b instruct provides a viable alternative, especially when labeled data is limited or unavailable. Future research could explore further optimization of fine-tuning strategies and investigate the generalizability of pre-trained models across different domains and tasks.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Zhenrui Yue, Huimin Zeng, Mengfei Lan, Heng Ji, and Dong Wang. 2023. Zero-and few-shot event detection via prompt-based meta learning. *arXiv preprint arXiv:2305.17373*.

JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21.

Transformers at SemEval-2024 Task 5: Legal Argument Reasoning Task in Civil Procedure using RoBERTa

Kriti Singhal[1], **Jatin Bedi**[2]
Computer Science and Engineering Department,
Thapar Institute of Engineering and Technology, India
[1]kritisinghal711@gmail.com, [2]jatin.bedi@thapar.edu

Abstract

Legal argument reasoning task in civil procedure is a new NLP task utilizing a dataset from the domain of the U.S. civil procedure. The task aims at identifying whether the solution to a question in the legal domain is correct or not. This paper describes the team "Transformers" submission to the Legal Argument Reasoning Task in Civil Procedure shared task at SemEval-2024 Task 5. We use a BERT-based architecture for the shared task. The highest F1-score score and accuracy achieved was 0.6172 and 0.6531 respectively. We secured the 13th rank in the Legal Argument Reasoning Task in Civil Procedure shared task.

1 Introduction

Mastering the art of arguing a legal case is essential for lawyers. This necessitates deep knowledge of the particular area of law along with advanced reasoning capabilities, including drawing similarities and differences. Researchers have made significant efforts towards setting the benchmark models for the new Natural Language Processing (NLP) problems in the domain of legal language understanding (Chalkidis et al., 2022).

The task, Legal Argument Reasoning Task in Civil Procedure[1] (Held and Habernal, 2024), organized at SemEval-2024 aimed at classifying the solution to a given problem as right or wrong.

Classifying an answer to a given question as correct or incorrect is a new NLP task. In particular, in the legal domain limited number of publicly available corpora exist. This contributes to added difficulty of this task (Fawei et al., 2016).

Recent advances in the field of NLP have addressed various issues, such as long texts and under-resourced domains. These include Long Short Term Memory (Hochreiter and Schmidhuber, 1997), and Gated Recurrent Units (Chung et al., 2014). But transformers (Vaswani et al., 2017) have taken the performance to new heights which were not possible earlier.

In the past, various efforts have been made to perform domain-specific adaption of different existing techniques and models. Some of these adaptions include SciBERT which was pre-trained for scientific texts, specifically in the bio-medical domain (Beltagy et al., 2019). Similarly, BioBERT was created with special emphasis on the bio-medical area (Lee et al., 2019).

In this paper, we discuss our use of a transformer-based model, RoBERTa, in the shared task of Legal Argument Reasoning Task in Civil Procedure at SemEval-2024.

2 Related Work

Researchers have used and explored various techniques in the past. In the work done by Beltagy et al. (2019); Lee et al. (2019), it was found that BERT-based architectures did not perform very well on problems that required specialized domain knowledge. Two possible solutions were found to address this issue. The first was to further pre-train BERT on domain-specific corpora, and the second possible solution was to pre-train BERT from scratch on domain-specific corpora (Chalkidis et al., 2020).

Lee et al. (2019) performed domain-adaption of BERT in the bio-medical domain. The experiment explored the effect of further pre-training BERT base for 470,000 steps on biomedical articles. The performance of the resulting model, BioBERT, was evaluated on biomedical datasets. This led to an improvement in performance when compared to BERT base.

Beltagy et al. (2019) proposed a family of BERT-based models, SciBERT, for scientific texts with a special focus on the bio-medical domain. Two approaches were followed for SciBERT, the first was further pre-training BERT base, and the second

[1]https://github.com/trusthlt/semeval24

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 969–972

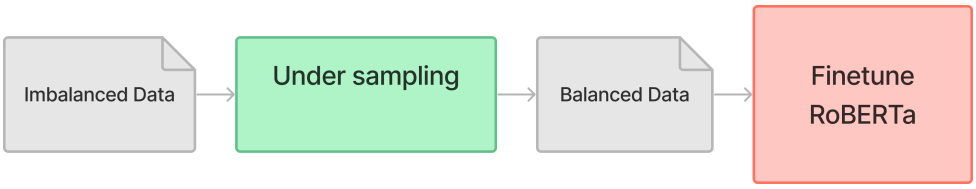


Figure 1: Proposed Methodology

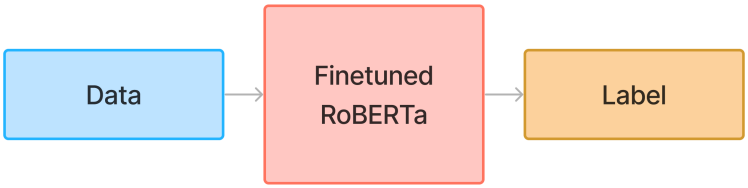


Figure 2: Label Generation for Unseen Data

approach was to pre-train BERT base on domain-specific corpora from scratch. In the second approach, random initialization of the model was performed, and a fresh new vocabulary was created. An improvement in performance was observed in the downstream tasks for both the approaches.

In the work carried out by Chalkidis et al. (2020), BERT domain adaption was performed for the legal domain. A systematic analysis was performed for the three available techniques. The first technique was to use BERT out of box, the second technique was to perform additional pre-training on BERT using domain-specific corpora, and the third approach was to perform pre-training from the start using domain-specific corpora.

3 Dataset Description

The dataset provided by the organizers was selected from the domain of the U.S. civil procedure and is based on a book aimed at law students.

In the training set, there are 666 instances, out of which 505 are labeled as 0 and 161 are labeled as 1. For each instance in the training data, there is a general introduction to a case, a question from that case, a possible argument solution along with a detailed analysis of why the argument is valid for that case. The test set, on the other hand, contains a question, answer and an explanation on the basis of which a label needs to be assigned to each instance. The assigned label will indicate whether or not the answer to the question is right or not.

4 Methodology

It was observed that in the dataset provided by the organizers, the number of instances in class 0 was 505, while the number of instances labeled as 1 was 161. Hence, in order to address the data imbalance, minority sampling was performed by randomly picking 161 instances from those labeled as class 0. This ensured that no bias existed in the trained model.

For identifying whether the answer to a given problem was correct or not, the RoBERTa Large model was employed. The RoBERTa model was designed by Facebook AI in 2019 (Liu et al., 2019). RoBERTa is a pre-trained transformer model which was trained in a self-supervised manner, i.e. only raw texts were used to train it without the involvement of human labeling.

While training the model, all the fields present in the training data, namely, question, answer, and analysis, were used to predict the provided label. The weighted Adam optimizer along with cross-entropy loss was used as the optimizer and the loss

Table 1: RoBERTa Performance Comparison

Model	F1-Score	Accuracy
RoBERTa Base	0.5511	0.6020
RoBERTa Large	0.6172	0.6531

function respectively. The learning of the optimizer was set at 1e-5. The RoBERTa model was trained for 100 epochs with the aforementioned parameters with a batch size of 8.

The training procedure has been summarised in Figure 1. The fine-tuned transformer was used to then predict the label for the unseen data as shown in Figure 2.

5 Results and Discussion

A BERT-based transformer, RoBERTa was discussed to perform categorization of an answer as right or wrong given a case, question, and a possible answer.

The data imbalance was handled by performing under sampling on the majority class instances in a random fashion. This was followed by fine-tuning the RoBERTa Large model for 100 epochs. After fine-tuning, the model achieved an F1 score of 0.5511 and an accuracy of 0.6020.

As shown in Table 1, the RoBERTa Base model performed better than RoBERTa Large, when fine-tuned for 100 epochs using the same methodology and hyper parameters. And it achieved an F1 score of 0.6172 and an accuracy of 0.6531.

Overall, we achieved the 13th rank in the Legal Argument Reasoning Task in Civil Procedure shared task at SemEval-2024 out of the 21 participating teams.

6 Conclusion and Future Work

Legal argument reasoning is a new NLP task, aimed at classifying a candidate answer as correct or incorrect given an introduction to the topic, a question and a candidate answer.

In this work, we describe our use of a BERT-based architecture, RoBERTa in the Legal Argument Reasoning Task in Civil Procedure shared task at SemEval-2024.

Ensembling techniques have shown promising results on various NLP tasks in different domains. Using an ensemble approach of different transformers may hence improve the performance. Transformers trained specifically with a focus on legal transformation such as Legal-BERT (Chalkidis et al., 2020) can improve the performance further.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Biralatei Fawei, Adam Wyner, and Jeff Pan. 2016. Passing a USA national bar exam: a first corpus for experimentation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3373–3378, Portorož, Slovenia. European Language Resources Association (ELRA).

Lena Held and Ivan Habernal. 2024. SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

YNU-HPCC at SemEval-2024 Task 7: Instruction Fine-tuning Models for Numerical Understanding and Generation

Kaiyuan Chen, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
chenkaiyuan@stu.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

This paper presents our systems for Task 7, Numeral-Aware Language Understanding and Generation of SemEval 2024. As participants of Task 7, we engage in all subtasks and implement corresponding systems for each subtask. All subtasks cover three aspects: Quantitative understanding (English), Reading Comprehension of the Numbers in the text (Chinese), and Numeral-Aware Headline Generation (English). Our approach explores employing instruction-tuned models (Flan-T5) or text-to-text models (T5) to accomplish the respective subtasks. We implement the instruction fine-tuning with or without demonstrations and employ similarity-based retrieval or manual methods to construct demonstrations for each example in instruction fine-tuning. Moreover, we reformulate the model's output into a chain-of-thought format with calculation expressions to enhance its reasoning performance for reasoning subtasks. The competitive results in all subtasks demonstrate the effectiveness of our systems.[1]

1 Introduction

In numerous domains, precise numerical information within text is decisive in decision-making and planning. Understanding and generating text- numbers would be beneficial for improving the model's performance on specific tasks. However, it poses challenges for existing models. Also, previous research indicates that current models struggle to properly represent textual numbers (Chen et al., 2023), often leading to inaccuracies.

Therefore, Task 7 of SemEval (Chen et al., 2024) 2024 focuses on numerically-aware language comprehension and generation, which includes quantitative understanding (Chen et al., 2023), reading comprehension of the numerals in text (Chen et al., 2021), and numeral-aware headline generation (Huang et al., 2023).

[1]Our code is available at https://github.com/ChenKy23/semeval2024-Task7

We explored all the subtasks of Task 7 and designed corresponding systems for each subtask. Our work and contributions can be summarized as follows:

For Subtask 1, We adopt the paradigm of instruction tuning (Chung et al., 2022) to complete all subtasks and explore manually crafting instances. Our results demonstrate that the instruction tuning model (Flan-T5) (Chung et al., 2022) performs comparably to the BERT model (Devlin et al., 2019) on the Quantitative Understanding task.

For Subtask 2, we utilized the mT5 model (Xue et al., 2021) pre-trained on multilingual corpus and the Randeng-T5 (Wang et al., 2022) pre-trained on Chinese corpus to implement the respective systems, as this task involves Chinese. Consistent with Task 1, we designed an instruction template for inputs and employed instruction fine-tuning.

For Subtask 3, similar instances are retrieved and organized into the input-output format to further enhance model's performance in in-context learning. Specifically, we structured the model's output into the format of chain-of-thought (CoT) (Wei et al., 2022) and inserted calculation expressions to improve model's reasoning performance. Our system achieved the highest scores of ROUGE, BERTScore, and MoverScore in headline generation while ranking 3^{th} in numerical reasoning task.

The remainder of this paper is organized as follows. In Section 2, we describe the related work of our system. The system overview is presented in Section 3. The details of the experiments, main results, and a conclusion are drawn in Sections 4, 5, and 6, respectively.

2 Related Work

In-context Learning. As a novel paradigm, in-context learning (Brown et al., 2020; Chung et al., 2022) has proven to enable large language models to adapt to unseen tasks with instruction and a few

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 973–979
June 20-21, 2024

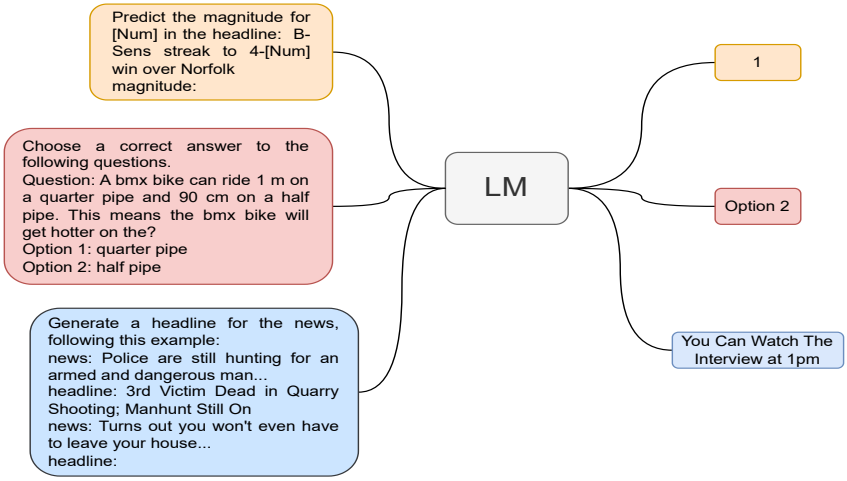


Figure 1: The application of instruction tuning across different tasks. In our system, LM represents either the Flan-T5 or T5 model. Different tasks employ different instruction templates, with or without demonstrations.

demonstrations, and it doesn't conduct any parameter updates. Furthermore, selecting semantically similar instances can further enhance the model's performance. (Liu et al., 2022; Rubin et al., 2022). Recent work has also applied in-context learning to fine-tuning small models (Fu et al., 2023).

Chain of Thoughts Prompt. CoT prompting (Wei et al., 2022; Kojima et al., 2022) is considered as a method to guide large language models (LLMs) in multi-step reasoning. In the numerical reasoning task of Subtask 3, the model's output can be reconstructed into a CoT format to enhance its reasoning performance. It's important to note that, unlike existing distillation methods (Fu et al., 2023), we don't use a LLMs to generate CoT rationales for each example. Instead, the original labels provided can be used to generate CoT rationales which is a more efficient way.

3 Overview of System

3.1 Instruction Tuning

The Instruction tuning models (Flan-T5 or T5) have developed strong generalization abilities through instruction fine-tuning across various tasks. Thus, appropriate instruction can lead to better model performance. While our system is not in a zero-shot setting, introducing instructions during the fine-tuning can enhance the model's performance. Therefore, we consider the input for all subtasks as the instruction template T concatenated with the query input x, i.e., $T + x$. In different tasks, T may have different meanings. The objective function

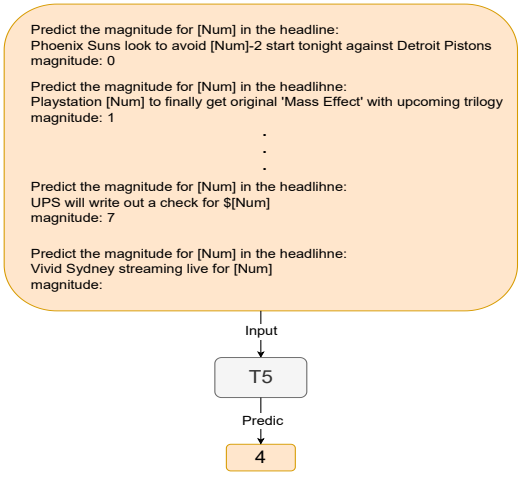


Figure 2: An example of instruction tuning with demonstrations from QP. According to different magnitudes, 8 instances can be selected manually.

for this process is as follows:

$$L_{instr} = \frac{1}{N}\sum_{i=1}^{N} CE\left(f\left(x_i, T\right), \widehat{y}_i\right) \quad (1)$$

where assume there are a total of N examples, x_i is i-th query input from the dataset, f is the output distribution function of models, CE represents the cross-entropy loss between predicted tokens and target tokens, and $\hat{y}_i$ denotes the tokens from the i-th gold label.

It's worth noting that the input-output formats vary for each subtask. We have designed distinct instruction formats for each task and employed instruction tuning to update the models across all tasks. Figure 1 illustrates how we employ instruction tuned models across various subtasks.

Operators	Expressions
$Copy(v)$	copy v from the news
$Trans(e)$	covert e into a number which represents v
$Paraphrase(v_0, n)$	paraphrase the form of v_0 to other representations in n «v_0/n=v_1»
$Round(v_0, v_1)$	hold v_0 digits after the decimal point of c, that is v_1
$Subtract(v_0, v_1)$	subtract v_0 from v_1 «v_0-v_1=v_2»
$Add(v_0, v_1)$	add v_0 and v_1 «v_0+v_1=v_2»;
$Span(s)$	select a span s from the article which represents 1;
$Divide(v_0, v_1)$	divide v_0 by v_1 «v_0/v_1=v_2»
$Multiply(v_0, v_1)$	multiply v_0 and v_1 «v_0*v_1=v_2»

Table 1: The 9 different operators can be translated into corresponding natural language expressions. Each expression might include an additional calculated number compared to the original operator.

3.2 Instruction Tuning with Demonstrations

The form of instruction tuning can be further expanded, where instructions can be subdivided into prompt P and a list of demonstrations D. P offers explicit guidance for the current task, while D provides the model with demonstrations of the input-output format. This paradigm has been recently referred to as in-context learning in related work (Brown et al., 2020; Chung et al., 2022). Therefore, the objective function for the extended instruction tuning can be expressed as follows:

$$L_{icl} = \frac{1}{N}\sum_{i=1}^{N} CE\left(f\left(x_i, P; D\right), \widehat{y}_i\right) \quad (2)$$

Based on the ways to select demonstrations, this method can be further categorized into manual and similarity-based instruction tuning.

Manual-based Instruction Tuning. This method is employed in Subtask 1 and 2. As subtask 1 involves various aspects, including QP, QNLI, QQA, the different demonstrations can be provided for each task. An example of how instruction is used for QP is shown in Figure 2. The manual selection of demonstrations are based on covering as many different results as possible.

Similarity-based Instruction Tuning. Using similarity-based retrieval for each input is more efficient and leads to better performance (Liu et al., 2022). We employed this method in the headline generation task for Subtask 3. First, pre-trained Sentence-BERT (Reimers and Gurevych, 2019) can be utilized as an encoder to map each news article x_i to a vector v_i. Then, with cosine similarity function F, the distances between v_i and all other vectors can be computed. Finally, the news article corresponding to the vector v_j, which has the closest distance, can be selected as a similar instance. The following function can represent this process:

$$v_i = S(x_i) \quad (3)$$

$$F(v_i, v_j) = \|v_i - v_j\|_2 (or \frac{v_i \cdot v_j}{\|v_i\|_2 \|v_j\|_2}) \quad (4)$$

$$v^i{}_{closest} = \underset{j \in \{1,2...N\} \cap j \neq i}{argmin} F(S(x_i), S(x_j)) \quad (5)$$

where N represents the total number of instances in the training set, S is the mapping function of Sentence-BERT, F denotes the cosine similarity function, and x_i represents a news article from the dataset.

3.3 Learning to Reasoning by CoT

As a part of Subtask 3, the numerical reasoning task requires deducing the numbers in masked headlines based on given news articles and approximately 20% of the questions involve reasoning and computation. Related operators can be categorized into 9 types, including $Copy$, Add, and others. Thus, directly predicting the numbers may be challenging. An example for NumHG (Huang et al., 2023) can be shown as follows:

Operations = Add(Subtract(5,3), Copy(3))

While this format of the execution process correctly deduces the results, it may not be very intuitive and does not provide the model with interpretable rationales. Therefore, it can be converted into a CoT format, which contains multiple immediate reasoning steps. The above example can be converted into the following CoT format:

First, subtract 3 from 5 «5-3=2»; Second, copy 3 from the news; Third, add 2 and 3 «2+3=5»;

We design corresponding natural language expressions for all 9 operators involved in the dataset and use the program to implement this process automatically. The complete correspondence between operators and natural language expressions

Notation	Model/Method	QP		QNLI					QQA	Score
		comment	headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
Original	BERT	**70.44**	57.46	64.40	**59.20**	72.29	60.42	99.91	53.20	67.17
	Link-BERT	68.81	55.70	59.94	56.85	73.43	59.01	99.91	54.14	65.97
	RoBERTa	60.46	58.03	60.15	57.64	**79.58**	58.77	98.93	51.96	65.69
	Flan-T5$_{inst}$	67.20	58.82	**77.73**	52.40	77.06	68.40	**99.94**	**59.25**	**70.10**
	Flan-T5$_{icl}$	66.68	**59.68**	74.74	52.07	76.85	**70.40**	**99.94**	56.17	69.57
Digit-based	BERT	65.38	54.74	57.86	56.46	71.36	60.11	99.11	53.75	64.85
	Link-BERT	63.76	55.41	59.54	**57.42**	73.63	60.17	99.73	53.44	65.39
	RoBERTa	**69.25**	57.65	59.40	56.69	**78.90**	62.38	99.91	54.34	67.31
	Flan-T5$_{inst}$	67.21	**58.56**	**74.70**	50.97	72.32	**68.40**	**100.00**	**58.02**	**68.77**

Table 2: The comparison between our system and previous work (Chen et al., 2023). The model used is Flan-T5-Base. $instr$ denotes fine-tuning with simple instruction prompts, while icl represents tuning with demonstrations. Refer to section 3.1 and 3.2 for more details. The *Original* refers to the inherent representation of numbers in the text, while *Digit-based* signifies the segmentation of numbers at the character level.

Model	Num Acc			ROUGE			BERTScore			MoreScore
	Overall	Copy	Reasoning	1	2	3	P	R	$F1$	
Flan-T5-Base$_{direct}$	64.247	68.828	55.904	43.64	20.21	39.16	45.56	45.08	45.33	58.84
Flan-T5-Base$_{instr}$	65.180	69.327	57.629	43.94	20.23	39.46	45.87	45.30	45.60	58.90
Flan-T5-Base$_{instr+truncate}$	65.196	69.426	57.493	44.08	20.40	39.50	46.03	45.56	45.80	58.96
Flan-T5-Base$_{icl}$	63.554	67.730	55.949	44.22	20.59	39.68	46.38	45.58	45.99	58.99
Flan-T5-XXL$_{int8_LoRA+instr}$	**70.686**	**75.262**	**62.352**	48.57	24.40	43.66	50.86	49.62	50.25	60.32
Flan-T5-XXL$_{int8_LoRA+icl}$	69.044	73.018	61.807	**48.90**	**24.71**	**44.22**	**51.58**	**50.10**	**50.85**	**60.55**

Table 3: The performance of different methods on the headline generation task of NumHG based on ROUGE, BERTScore, and Num Acc. The $direct$ indicates directly fine-tuning the model without instruction, and $truncate$ signifies truncating the input to a length of 512.

is shown in Table 1. Furthermore, expressions can be inserted for each computational operation. An external calculator (Cobbe et al., 2021) can be used for result correction. For the mentioned example, if the model output is *5-3=1*, the external calculator will correct it to the right result, which is *5-3=2*. Subsequently, string matching replaces the incorrect numerical values in the sequence.

4 Experiment Details

Datasets. Subtask 1 utilizes Quantitative 101 (Chen et al., 2023) as the dataset, encompassing three aspects: QP (Chen et al., 2019), QQA (Mishra et al., 2022), and QNLI (Ravichander et al., 2019); Subtask 2 utilizes NQuAD (Chen et al., 2021), which is a Chinese machine reading comprehension task; NumHG (Huang et al., 2023) is used in Subtask 3, which comprises over 27K annotated numeral-rich news articles and can be further divided into headline generation and numerical reasoning.

Model Selection. For Subtasks 1 and 3, Flan-T5 (Chung et al., 2022) is utilized. For Subtask 2, we employ mT5-Small (Xue et al., 2021) and Randeng-T5-77M (Wang et al., 2022). Specifically, for Subtask 3, we experimented with Flan-T5 models ranging from Base to XXL sizes. For Flan-T5-XL and Flan-T5-XXL, we applied 8-bit quantization (Dettmersλ et al., 2022) and performed parameter-efficient tuning using LoRA (Hu et al., 2022). The *all-mpnet-base-v2* [2] can be utilized as encoder to map the text to vector.

Hyper-Parameter Selection. Adamw (Loshchilov and Hutter, 2017) is employed as the optimizer. In Subtask 1, The learning rate for all four QNLI tasks and QQA is set to 5e-7. For the QP task, the learning rate is set to 3e-5. Unless specified, the learning rates, dropout and warm-up rates for remaining tasks are set to 5e-5, 1e-2 and 0.1, respectively. We also applie the PEFT[3] library for parameter-efficient tuning.

Evaluation Metrics. Quantitative-101 Score (Huang et al., 2023) is used for ranking the overall performance in Subtask 1, while Accuracy is used to evaluate Subtask 2 and the numerical reasoning task of Subtask 3. For the numerical reasoning task, based on whether the reasoning question involves calculation, they can be further categorized into *simple* and *complex*. ROUGE (Lin, 2004), BERTScore (Zhang et al., 2020), and MoverScore (Zhao et al., 2019) are used to evaluate the result of the headline generation task of Subtask 3 and nu-

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[3] https://github.com/huggingface/peft

Method/Model	Accuracy
BERT Embedding Similarity	57.30
Vanilla BERT	66.41
BERT-BiGRU	67.15
BERT-CNN	63.92
NEMo	69.95
Randeng-T5-77M	**89.71**
mT5-Small	88.82
mT5-Base$_{LoRA}$	80.42

Table 4: The comparative results on NQuAD. Some results come from previous work (Chen et al., 2021). Evaluation is based on accuracy (%).

Model	ROUGE		
	1	2	3
Flan-T5-Base$_{instr}$	44.67	20.90	40.27
Flan-T5-Large$_{instr}$	47.07	22.58	42.04
Flan-T5-XL$_{int8_LoRA+instr}$	48.36	23.69	43.45
Flan-T5-XXL$_{int8_LoRA+instr}$	**49.58**	**24.98**	**44.69**
Flan-T5-Base$_{icl}$	44.88	21.02	40.57
Flan-T5-XXL$_{int8_LoRA+icl}$	**49.60**	**25.27**	**45.01**

Table 5: The results of models at different scales on the dev set of headline generation.

merical accuracy in headlines is also considered.

5 Main Result and Analysis

Comparison Results on Quantitative 101 As results shown in Table 2, despite the distinct ways to handling queries, the instruction tuning Flan-T5 remains comparable to BERT. Notably, our system performs superior on QNLI and QQA, which have smaller datasets. The introduction of manual demonstrations (Sec 3.2) don't lead to improvement in instruction fine-tuning. This may be associated with the manual selection of examples and hyperparameters. Furthermore, in contrast to the *Digit-based* notations, utilizing the *Original* notation for numbers performs better.

Comparison Results on NQuAD As shown in Table 4, it can be observed that the T5 tuning by instruction outperformed the BERT significantly. Both mT5 and Randeng-T5 are pre-trained on multilingual or Chinese corpus, which can enhance their capability to address Chinese-related tasks effectively. Additionally, Randeng-T5, which is based on Chinese corpus, is superior to mT5. However, enlarging the model scale seemed to lead to decreased accuracy on this task.

Comparison Results on NumHG Tables 5 and 6 show that larger models perform better on both headline generation and numerical reasoning.

Table 3 shows that instruction fine-tuning indeed leads to better performance on text generation compared to direct fine-tuning, which means instructions providing proper guidance to Flan-T5. Interestingly, the introduction of similar demonstrations further enhances the model's performance on text generation evaluation metrics but comes at the cost of lower numerical accuracy, which can be observed both in the model of Base and XXL.

Method	Num Acc		
	Total	Simple	Complex
Flan-T5-Base$_{ans_only}$	88.691	94.205	61.125
Flan-T5-Base$_{operator}$	88.753	94.548	59.780
Flan-T5-Base$_{cot}$	88.509	94.279	59.658
Flan-T5-Base$_{cot+cal}$	88.936	94.279	62.225
Flan-T5-XXL$_{int8_LoRA+operator}$	93.704	97.164	76.406
Flan-T5-XXL$_{int8_LoRA+cot}$	94.010	**97.359**	77.262
Flan-T5-XXL$_{int8_LoRA+cot+cal}$	**94.173**	**97.359**	**78.240**

Table 6: The performance of different methods on the numerical reasoning task. The *cot* is the method proposed in Sec 3.3, and *cal* denotes using an external calculator (Cobbe et al., 2021) for result correction.

As for numerical reasoning, the CoT method leads to better performance on answering *Complex* questions compared to other methods and external calculator correction further amplifies this advantage, as shown in Table 6. For both Base and XXL models, the CoT method under external calculator correction achieved the best performance. However, due to the relatively limited capabilities of smaller models, the performance boost on *Complex* tasks don't contribute significantly to the overall performance for the Base model.

6 Conclusion

During Task 7 of SemEval2024, we participated in all the subtasks and implemented the corresponding systems by instruction fine-tuning. We utilized instruction fine-tuning with demonstrations to expand its format. We also reformulated the output in the form of a chain of thought to improve the model's reasoning abilities. Our approach proved to be highly effective by outstanding performance across all the subtasks. In future work, we plan to further explore the impact of varying instance quantities, instruction templates, and model sizes on the results.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051. We would like to thank the anonymous reviewers for their constructive comments.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Nquad: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 2925–2929, New York, NY, USA. Association for Computing Machinery.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmersλ, Mike Lewis, Younes Belkada, Luke Zettlemoyer, Hugging Face, and ENS Paris-Saclay. 2022. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2655–2671. Association for Computational Linguistics.

Junjie Wang, Yuxiang Zhang, Lin Zhang, Ping Yang, Xinyu Gao, Ziwei Wu, Xiaoqun Dong, Junqing He, Jianheng Zhuo, Qi Yang, Yongfeng Huang, Xiayu Li, Yanghan Wu, Junyu Lu, Xinyu Zhu, Weifeng Chen, Ting Han, Kunhao Pan, Rui Wang, Hao Wang, Xiaojun Wu, Zhongshen Zeng, Chongpei Chen, Ruyi Gan, and Jiaxing Zhang. 2022. Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence. *CoRR*, abs/2209.02970.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *NAACL-HLT*, pages 483–498. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

CAILMD-23 at SemEval-2024 Task 1: Multilingual Evaluation of Semantic Textual Relatedness

Srushti Sonavane[1] , **Sharvi Endait**[1] , **Ridhima Sinare**[1] , **Pritika Rohera**[1], **Advait Naik**[1], and **Dipali Kadam**[1]
Pune Institute of Computer Technology, Pune[1]

Abstract

The explosive growth of online content demands robust Natural Language Processing (NLP) techniques that can capture nuanced meanings and cultural context across diverse languages. Semantic Textual Relatedness (STR) goes beyond superficial word overlap, considering linguistic elements and non-linguistic factors like topic, sentiment, and perspective. Despite its pivotal role, prior NLP research has predominantly focused on English, limiting its applicability across languages. Addressing this gap, our paper dives into capturing deeper connections between sentences beyond simple word overlap. Going beyond English-centric NLP research, we explore STR in Marathi, Hindi, Spanish, and English, unlocking the potential for information retrieval, machine translation, and more. Leveraging the SemEval-2024 shared task, we explore various language models across three learning paradigms: supervised, unsupervised, and cross-lingual. Our comprehensive methodology gains promising results, demonstrating the effectiveness of our approach. This work aims to not only showcase our achievements but also inspire further research in multilingual STR, particularly for low-resourced languages. (Ousidhoum et al., 2024b)

Keywords: Natural Language Processing, Semantic Textual Relatedness, Sentence Transformers, supervised learning, unsupervised learning, cross-lingual.

1 Introduction

The ever-increasing diversity of online content demands robust Natural Language Processing (NLP) techniques that can grasp the nuances of meaning across diverse languages. Semantic Textual Relatedness (STR) plays a crucial role in achieving this goal by delving beyond superficial lexical similarity and capturing the deeper connections between sentences. Unlike semantic similarity, which focuses solely on the taxonomic overlap of words, STR encompasses both linguistic elements and non-linguistic factors like the topic, point of view, and period. This richer understanding unlocks significant potential in various NLP tasks, regardless of the user's native language. Imagine searching for information online in your native language and receiving results that truly understand your intent, and not just match keywords. STR holds the key to unlocking this dream, bridging the language gap, and fostering true multilingual communication.

Despite the recognized importance of Semantic Textual Relatedness (STR) for multilingual communication, most prior NLP research has focused on semantic similarity within English due to limitations in labeled data for diverse languages (Abdalla et al., 2023). This narrow focus restricts the potential of STR applications like information retrieval across languages with different cultural contexts or machine translation that accurately captures nuances beyond direct word equivalents. Existing relatedness methods primarily target English (Hasan and Halliday, 1976), with limited exploration in languages like German, Chinese, and Japanese (Zesch et al., 2007) (Li et al., 2005) (De Saeger et al., 2010). This highlights a critical gap in Natural Language Processing (NLP): accurately measuring semantic relatedness across diverse languages.

The identified gap in multilingual STR research, with its limitations in diverse language applications, demands innovative solutions. This paper dives into the exciting realm of bridging this gap through multilingual Semantic Textual Relatedness (STR). Specifically, we explore methods to capture the semantic connections between texts in languages like English, Marathi, Hindi, and Spanish.

Our research focuses on the SemEval-2024 shared task, which provides three tracks to evaluate STR techniques:

1. Supervised Learning: This track focuses on building systems trained on the provided labeled datasets.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 980–985

2. Unsupervised Learning: Here, the challenge lies in developing systems that learn semantic relationships without relying on any labeled data.

3. Cross-lingual Learning: This track pushes the boundaries by requiring systems to leverage knowledge from labeled data in a source language (Track A) to address a target language with limited resources.

For each track, we present a comprehensive methodology, employing diverse language models and rigorously analyzing their performance. This allows us to identify the most effective approaches for each challenge. Notably, our submissions achieved promising scores on several tracks, demonstrating the strength and potential of our proposed methods.

Looking beyond our achievements, this work aims to inspire further exploration of multilingual STR, particularly for under-resourced languages. We believe that larger datasets and broader language coverage hold immense potential to benefit the NLP community, unlocking the true potential of language understanding and empowering communication across diverse cultures.

2 Related Work

Semantic textual relations (STR) play an important role in natural language processing (NLP), which aims to identify the degree of semantic similarity between text groups. It forms the backbone of various NLP tasks such as information retrieval, question answering, and paraphrase detection, necessitating the assessment of similarity between sentences, phrases, or documents.

Historically, detailed STR research from the 1900s through the 2000s relied heavily on statistical methods heavily dependent on lexical databases like WordNet. However, these methods suffered from a lack of real-world knowledge integration (Gabrilovich et al., 2007). Classified translation emerged with developments such as GloVe (Pennington et al., 2014), Word2Vec, and FastText, which enabled text to be converted into word input. The current methods require converting corpora into words or sentence-embedded forms and computing connectivity scores. Notably, large language models (LLMs) such as Sentence-BERT often use Cosine Similarity in embedded sentences to measure relatedness (Gunawan et al., 2018) (Reimers and Gurevych, 2019).

Previous methodologies have delved into both knowledge-based (ontology, classification) and corpus-based (unsupervised learning) approaches. For example, (Siblini and Kosseim, 2017) examined three approaches: semantic linkage, classification similarity, and hybrid approaches. Notably, the multilingual approach of (Hasan and Halliday, 1976), improved by 47%, confirming the potential of emphasis on the use of multilingual strategies. Furthermore, studies on less resourceful African languages highlight the need for different data types and methodologies (Delil and Kuyumcu, 2023).

A significant challenge in STR lies in the scarcity of huge-scale, promising datasets for education and assessment. Initiatives like SemEval play a pivotal role in addressing this gap through dedicated shared tasks focused on STR (Abdalla et al., 2023). These collaborative efforts foster the improvement and evaluation of STR models throughout diverse linguistic landscapes and domain names.

The current advent of the STR-2022 dataset by (Abdalla et al., 2023) marks a significant leap forward in STR studies. This annotated dataset, comprising sentence pairs with relatedness scores, serves as a precious aid for schooling and evaluating STR fashions. Covering various domains and languages, it displays the multilingual nature of STR studies (Abdalla et al., 2023).

Moreover, STR-2022 addresses biases and perceptions in relatedness judgments. Through meticulous curation, it aims to mitigate biases and ensure annotation quality, thereby fostering fair evaluations and robust model development. Additionally, the dataset highlights the relative nature of relatedness ratings, emphasizing the significance of context and assignment-precise thresholds in decoding similarity measures (Abdalla et al., 2023).

3 System Description

In this section, we aim to outline our system's components for assessing semantic textual relatedness across different datasets: a) labeled datasets using supervised learning, b) unlabeled datasets employing unsupervised learning, and c) cross-lingual datasets. We'll detail the utilized data, the models employed in each track, and the results obtained from training these models on respective datasets.

3.1 Data Collection

We utilized the SemRel2024 Dataset (Ousidhoum et al., 2024a) for training and evaluating our final results. This comprehensive dataset consists of semantic textual relatedness data across 14 diverse languages, including Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu. From this array of languages, we focused on English, Hindi, Marathi, and Spanish datasets for our analysis.

Each entry in the dataset comprises a sentence pair along with its corresponding semantic similarity score. This score ranges from 0 to 1, where 0 signifies no similarity between the sentences, while 1 indicates complete similarity.

For the supervised track (Track A), we concentrated on English and Marathi datasets. In the unsupervised track (Track B), our attention was on English and Hindi datasets. Lastly, for Track C, we employed English and Hindi datasets, utilizing Spanish and English as their language training bases, respectively.

Language	Train	Dev	Test
English	5500	250	2500
Hindi	-	288	968
Marathi	1155	293	298
Spanish	1592	140	600

Table 1: Distribution of dataset for Training, Development, and Testing

3.2 Experiments

3.2.1 Track A:

The SemRel2024 English and Hindi datasets were initially trained on baseline models such as Support Vector Regression and XGBoost. However, we additionally adapted the sentence-transformer-based models, such as all-mpnet-base-v2 [1] and marathi-sentence-bert-nli [2] by L3Cube for English and Marathi respectively. This was done to compensate for the smaller size of the corpora available, as these sentence transformer models are trained on a larger data size initially, and this would be efficient to understand not only the n-gram sequences but also the context of the sentences that are being compared.

[1]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[2]https://huggingface.co/l3cube-pune/marathi-sentence-bert-nli

Preprocessing and Feature vectorization were done using Term-frequency and inverse-document-frequency (TF-IDF) to generate vectors and preprocess the models SVR and XGBoost. Term-frequency, Inverse-Document-Frequency (TF-IDF) is a numerical statistic that determines how important a word is in a given document or a piece of textual content. This is done by multiplying two metrics: How many times a word appears in a document Inverse document frequency of the word across a set of documents. This score for word in the document d from document D is calculated as follows:

$$tfidf(t, d, D) = tf(t, d).idf(t, D) \quad (1)$$

$$tf(t, d) = log(1 + freq(t, d)) \quad (2)$$

$$idf(t, D) = \log\left(\frac{N}{\text{count}(d \in D : t \in d)}\right) \quad (3)$$

(Chen et al., 2020).

Support vector regression (SVR) can be used for calculating semantic similarity scores between sentences. It's a supervised learning algorithm that can model the relationship between input features, (here in this case the sentences) and the output labels (semantic similarity scores in this case).

XGBoost can be used for semantic similarity score calculation between sentences as it is a powerful gradient-boosting algorithm, and it can be applied to various supervised learning tasks. This is capable of handling complex non-linear relationships between features and labels, and it's robust against overfitting.

All-mpnet-base-v2 (All- Massively Parallel Multilingual Transformer) is a sentence encoder model, given an input text, it gives a vector that collects the semantic information. The sentence vector is used for tasks such as clustering or sentence similarity tasks. This is a sentence-transformers model and it maps sentences & paragraphs to a 768-dimensional dense vector space. This is trained on the SemRel2024 dataset and additionally, it is capable of capturing long-range dependencies, and it leads to higher performance on text classification, NER, and question answering.

Marathi-Sentence-Bert-Nli (Joshi et al., 2022) is a Marathi sentence transformer model that has been trained on synthetic STS and NLI datasets. These are fine-tuned on MahaBERT, a BERT-based model that is fine-tuned on a large Marathi corpora.

3.2.2 Track B:

For track B, the unsupervised track, the SemRel2024 dev set, and the test set were used for testing the model selected. The languages were English and Hindi, for which Track-B dev and test sets were used. The models used for this were BERT-based uncased and Hindi-Bert v2 (Joshi, 2022) accordingly.

Hindi-BERT-v2 was roughly trained on 1.8 B tokens. Compared to general-purpose language models, this monolingual model is optimized to understand and process Hindi text effectively. Due to the larger corpus it has been trained upon this has been an accurate model to obtain results from.

BERT-based-uncased (Bidirectional Encoder Representations from Transformers) is trained on uncased text. BERT is based on the transformer architecture that relies on self-attention mechanisms to capture relationships between words in a sequence, enabling effective modeling of long-range dependencies in text data.

Algorithm:

1. The BERT model (English/Hindi) is initialized.
2. Sentence embeddings for sentence 1 are calculated.
3. Sentence embeddings for sentence 2 are calculated.
4. Calculate the cosine similarity scores of the embeddings.

3.2.3 Track C:

Cross-linguistic track: The English and Spanish SemRel2024 training datasets were used for training data in languages Hindi and English respectively. The dataset first underwent translation using the deep translation API. By translating the dataset from English to Hindi and Spanish to English, the training dataset for that language was available and was used for testing the development set and the test set of the SemRel 2024 dataset. The models used for training the dataset were the "all-mpnet-base-v2" sentence transformer and "hindi-sentence-bert-nli" [3] by L3cube. These 2 sentence transformers are discussed in Track A, above, however, this went through an additional translation pipeline before that.

[3]https://huggingface.co/l3cube-pune/hindi-sentence-similarity-sbert

1. Translate sentences from Language 1 to Language 2 using an appropriate translation service or tool.
2. Initialize the model for the task, such as sentence similarity or classification.
3. Encode Sentence1 into a numerical representation using the initialized model. This involves converting the text input into a format suitable for processing by the model, typically through tokenization and embedding.
4. Similarly, encode Sentence2 into a numerical representation using the same BERT model.
5. Train the initialized model on the provided dataset. This step involves feeding the encoded sentence pairs into the model and adjusting the model's parameters to minimize a predefined loss function, typically using techniques like backpropagation and gradient descent.
6. Evaluate the performance of the trained model on a separate evaluation dataset or through cross-validation. This step aims to assess the model's ability to generalize to unseen data and its overall effectiveness in the task of interest, such as sentence similarity or classification.
7. If necessary, repeat steps 3 to 6 for all pairs of sentences in the dataset. This process ensures that the model learns from a diverse range of examples and improves its performance across different input scenarios.

4 Experimental Setup

The dimensions of the dataset splits are summarized in Table 1, indicating the number of samples allocated for training, development, and testing across different languages. The experimental setup encompassed preprocessing procedures, leveraging Hugging Face Transformers for model access, NumPy for array operations, Pandas for data manipulation, Sentence Transformers for sentence embeddings, and NLTK for various NLP tasks. Evaluation measures such as the F1 score, accuracy, and recall were employed to comprehensively assess the performance of the models across correlation, classification accuracy, and retrieval quality aspects.

5 Results

Tables 2, 3, and 4 present the performance results for the three setups (supervised, unsupervised, and cross-lingual) in our experiments. Each table summarizes the F1 score, accuracy, and recall achieved by various models for each language.

Sr.No.	Language	Model Name	F1	Accuracy	Recall
1	English	BERT-base-nli	0.87	0.876	0.84
2	English	SVR	0.59	0.55	0.65
3	English	XG-Boost	0.79	0.82	0.76
4	Marathi	Marathi-NLI	0.83	0.81	0.90
5	Marathi	SVR	0.63	0.61	0.60
6	Marathi	XGBoost	0.66	0.67	0.70

Table 2: Performance Metrics for Track A

This table shows the performance of models in the supervised learning setup, where labeled data was available for training. BERT-based models ("BERT-nli" and "Marathi-nli") consistently outperform other models (SVR, XGBoost) in both English and Marathi, achieving significantly higher correlation scores (0.823 and 0.871, respectively). Interestingly, the Marathi-specific nli model even surpasses the multilingual BERT performance in Marathi, suggesting the benefit of language-specific models.

Sr. No.	Language	Model Name	F1	Accuracy	Recall
1	English	sentence-t5	0.66	0.49	0.49
2	English	BERT based uncased	0.85	0.86	0.80
3	Hindi	Indic-BERT	0.66	0.50	0.50
4	Hindi	hindi-bert-v2	0.66	0.74	0.50

Table 3: Performance Metrics for Track B

This table presents the results for the unsupervised learning setup, where models were trained without relying on labeled data. In English, the BERT-based model ("BERT-base-uncased" [4]) outperforms the pre-trained Sentence-T5 [5] model, possibly due to its larger size and fine-tuning on relevant NLP tasks. In Hindi, while both Indic-BERT(Kakwani et al., 2020) and hindi-bert-v2 (Joshi et al., 2022) have similar F1 scores (around 66%), the latter achieves a significantly higher correlation coefficient (0.796). This indicates that hindi-bert-v2 [6] captures semantic relatedness more effectively despite similar overall accuracy.

[4]https://huggingface.co/google-bert/bert-base-uncased

[5]https://huggingface.co/sentence-transformers/sentence-t5-base

[6]https://huggingface.co/l3cube-pune/hindi-bert-v2

Sr. No.	Language	Model Name	F1	Accuracy	Recall
1	Spanish to English	all-mpnet-base-v2	0.82	0.82	0.81
2	English to Hindi	hindi-sentence-bert-nli	0.71	0.77	0.92

Table 4: Performance Metrics for Track C

This table shows the performance of models in the cross-lingual learning setup, where the goal was to assess semantic relatedness across different languages. Both models used ("all-mpnet-base-v2" for Spanish-to-English and "hindi-sentence-bert-nli" for English-to-Hindi) achieve worthy correlation scores (0.786 and 0.809, respectively) demonstrating the potential of cross-lingual approaches.

Table 5 represents the results of the development phase.

Track	Language	Sp. Corr Coeff
A	English	0.812
	Marathi	0.855
B	Hindi	0.819
	English	0.825
C	Hindi	0.825
	Englsih	0.790

Table 5: Development Phase Results

6 Conclusion

In this paper, we presented a comparative analysis of systems for the Semantic Textual Relatedness (STR) task at SemEval-2024 Task 1. Our approaches, primarily based on language-specific transformer models, achieved top scores on several tracks, including 1st place in Unsupervised Learning for Hindi. Notably, we did not utilize any external datasets, highlighting the effectiveness of our approach despite potential variations in pre-trained model training data.

Prior research in STR has largely focused on English due to limited labeled data for diverse languages. This restricts the true prospect of STR applications like multilingual information retrieval and machine translation. We addressed this gap by exploring solutions for STR in English, Marathi, Hindi, and Spanish. We aim to inspire further research on multilingual STR, particularly for low-resourced languages. We believe larger datasets and broader language coverage hold immense potential for multilingual NLP, unlocking a deeper understanding and empowering cross-cultural communication.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2023. What makes sentences semantically related: A textual relatedness dataset and empirical study.

Weilong Chen, Xin Yuan, Sai Zhang, Jiehui Wu, Yanru Zhang, and Yan Wang. 2020. Ferryman at semeval-2020 task 3: bert with tfidf-weighting for predicting the effect of context in word similarity. In *Proceedings of the fourteenth workshop on semantic evaluation*, pages 281–285.

Stijn De Saeger, Kow Kuroda, Masaki Murata, Kentaro Torisawa, et al. 2010. A bayesian method for robust estimation of distributional similarities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 247–256.

Selman Delil and Birol Kuyumcu. 2023. Sefamerve at semeval-2023 task 12: Semantic evaluation of rarely studied languages. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 512–516.

Evgeniy Gabrilovich, Shaul Markovitch, et al. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611.

Dani Gunawan, CA Sembiring, and Mohammad Andri Budiman. 2018. The implementation of cosine similarity to calculate text relevance between two documents. In *Journal of physics: conference series*, volume 978, page 012120. IOP Publishing.

Ruqaiya Hasan and Michael AK Halliday. 1976. Cohesion in english. *London, 1976; Martin JR*.

Ananya Joshi, Aditi Kajale, Janhavi Gadre, Samruddhi Deode, and Raviraj Joshi. 2022. L3cube-mahasbert and hindsbert: Sentence bert models and benchmarking bert sentence representations for hindi and marathi.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Wanyin Li, Qin Lu, and Ruifeng Xu. 2005. Similarity based chinese synonym collocation extraction. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 10, Number 1, March 2005*, pages 123–144.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Reda Siblini and Leila Kosseim. 2017. Clac: Semantic relatedness of words and phrases.

Torsten Zesch, Iryna Gurevych, and Max Mühlhäuser. 2007. Comparing wikipedia and german wordnet by evaluating semantic relatedness on multiple datasets. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 205–208.

SEME at SemEval-2024 Task 2: Comparing Masked and Generative Language Models on Natural Language Inference for Clinical Trials

Mathilde Aguiar, Pierre Zweigenbaum, Nona Naderi
Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique,
91405, Orsay, France
{mathilde.aguiar, pierre.zweigenbaum, nona.naderi}@lisn.fr

Abstract

This paper describes our submission to Task 2 of SemEval-2024: Safe Biomedical Natural Language Inference for Clinical Trials. The Multi-evidence Natural Language Inference for Clinical Trial Data (NLI4CT) consists of a Textual Entailment (TE) task focused on the evaluation of the consistency and faithfulness of Natural Language Inference (NLI) models applied to Clinical Trial Reports (CTR). We test 2 distinct approaches, one based on finetuning and ensembling Masked Language Models and the other based on prompting Large Language Models using templates, in particular, using Chain-Of-Thought and Contrastive Chain-Of-Thought. Prompting Flan-T5-large in a 2-shot setting leads to our best system that achieves 0.57 F1 score, 0.64 Faithfulness, and 0.56 Consistency.

1 Introduction

The digitization of medical documents allows the development of tools using various NLP techniques. In the case of Clinical Trial Reports (CTR), these tools can facilitate recruiting patients to participate in a trial or help researchers keep up to date with the literature. Natural Language Inference (NLI) is particularly useful in detecting the relationship between a CTR and a statement. For instance, it can be used for patient-trial matching.

Task 2 of SemEval 2024 defines a Textual Entailment (TE) task applied to English breast cancer CTRs. A submitted system must perform a binary classification based on a CTR and a given statement, using the labels *entailment* or *contradiction*. In addition to the traditional F1-measure for Textual Entailment, the submitted systems are evaluated on 2 strong metrics: Faithfulness and Consistency.

In this paper, we first introduce the task and some related work in Sec. 2. Sec. 3 describes our proposed approaches, while Sec. 4 gives further details about the experimental setup. Sec. 5 presents the results and comparative analysis of methods, and Sec. 6 sums up our work done and provides ideas for future work.

2 Background

2.1 Corpus and task description

The NLI4CT (Jullien et al., 2024) corpus consists of a collection of breast cancer Clinical Trial Reports (CTR) taken from clinicaltrials.gov. The documents are exclusively written in English. These CTRs are structured with the following sections: *Intervention* section describes what treatment is going to be applied during the trial. *Eligibility* section consists of a set of inclusion and exclusion criteria that a test subject must comply with. *Results* section displays the outcome measures. Finally, *Adverse Events* section describes the side effects and symptoms observed during the trial. In NLI4CT there are two types of instances: *single*, where only 1 CTR is involved to perform the inference, and *comparison* where 2 CTRs need to be compared.

The task's objective is to perform Natural Language Inference on these clinical trials. A premise consists of a section of a CTR (or two CTRs if it is a comparison), and a statement is a single sentence. The model should predict whether the premise entails or contradicts the statement. To tackle the NLI4CT task, the model must perform several kinds of inference, such as quantitative, common-sense, and medical reasoning (see Fig. 2). The inference relationship can be predicted using the evidence, sentences where clues are contained, that are in one of the sections of a CTR. Evidence is provided only in the development and training sets. The dataset is balanced with half of the instances labeled as *entailment* and the other half as *contradiction* in the train and development subsets.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 986–996

2.2 Related work

A previous edition of the NLI4CT task was run as SemEval 2023 Task 7 (Jullien et al., 2023a). It was composed of 2 subtasks: an NLI classification task and an information retrieval task of evidence selection to support the predicted label. The training and development sets were the same as the present edition. For the first subtask, the task overview paper (Jullien et al., 2023b) reports both generative and discriminative approaches for the submitted systems. Over the past few years, we have seen the fast-paced development of Large Language Models (LLMs) and their increased capabilities in addressing both generative and discriminative tasks. Even general-domain LLMs like Flan-T5-xxl in Kanakarajan and Sankarasubbu (2023) and GPT-3.5 in Pahwa and Pahwa (2023) have been achieving competitive performance on domain-specific tasks for the 2023 edition of the NLI4CT task.

3 System overview

To address the NLI4CT task, we tested 2 main approaches: the first uses Pretrained Masked Language Models (MLM), and the second uses generative Large Language Models. We wanted to compare the ability of these two kinds of architectures to solve the same task, in particular in terms of consistency and faithfulness.

3.1 Finetuning pretrained masked language models

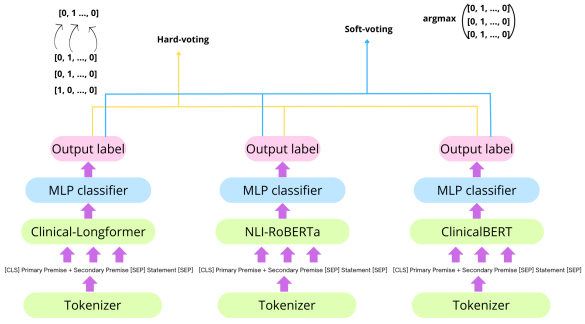


Figure 1: MLM ensemble architecture overview.

Our first system is based on finetuning and ensembling multiple MLMs on the task data (see an example in Fig. 1). We first finetune each model using the train and development splits of NLI4CT. We evaluate each finetuned model on the test set. We perform experiments with two ensembling methods: hard-voting and soft-voting. The hard-voting method consists of selecting the label y that gets the majority of votes across the predictions of each model j, defined as follows:

$$\tilde{y} = \underset{y}{\text{argmax}} \sum_{j=1}^{N} \mathbb{1}(\tilde{y}_j = y)$$

Soft-voting is computed by using the argmax of probabilities P_j from each model j for a given label y:

$$\tilde{y} = \underset{y}{\text{argmax}} \sum_{j=1}^{N} P_j(y)$$

3.2 Prompting generative large language models

We designed a set of prompts that rely on the following techniques:

1. A simple prompt instructing the model to perform Textual Entailment, giving the statement and a premise composed of the whole section where the evidence comes from. We took inspiration from the instruction templates found in the Flan-Muffin dataset[1] that (Lou et al., 2024) used to instruction-tune the Flan-T5 models (Chung et al., 2022). The template starts with optional demonstrations that instantiate this prompt with n training or development examples in n-shot settings:

[Demonstrations] [Premise] [Statement] Based on this premise, is the hypothesis true? OPTIONS: -'Yes' -'No'

2. Using the concept of Chain-Of-Thought (Wei et al., 2022) that decomposes the reasoning behind a given example; we insert the premise sentences that are the actual evidence used to infer an entailment or a contradiction in the demonstrations. See C.2 for a detailed example.

3. We tested the related Contrastive Chain-Of-Thought (CCOT) (Chia et al., 2023) technique that gives both one correct and one incorrect explanation in addition to the original template. In our case, we inserted premise sentences that were not actual evidence. See C.3 for an example. CCOT is inspired by how humans learn from positive and negative examples and aims to reduce reasoning errors by indicating what mistakes to avoid.

For the demonstrations, we tried three few-shot settings: zero-shot (ZS: no demonstration, only for the first template), 1-shot, and 2-shot. See Appendix C for detailed examples of the prompts.

[1] https://huggingface.co/datasets/causal-lm/flan-muffin

4 Experimental setup

4.1 Data pre-processing

We used the NLI4CT train and development splits published by BigBio on HuggingFace[2] and enriched them with new columns: primary and secondary evidence and premises from the JSON files provided by the organizers. We used this dataset to build our prompts (see Sec. 3.2). We shuffled the train and dev sets and selected random instances to include as demonstrations in our 1 and 2-shot settings.

4.1.1 Ensembling MLMs

We used Masked Language Models that are pretrained on general domain data or clinical data. For the general domain, we selected NLI-RoBERTa[3] (Reimers and Gurevych, 2019) from Sentence Transformers, which has been previously finetuned for NLI using SICK (Marelli et al., 2014) and STS benchmark (Cer et al., 2017). For the clinical pretrained models we use Clinical-Longformer[4] (Li et al., 2023), which can handle a context window up to 4096 tokens, and ClinicalBERT[5] (Wang et al., 2023) which has been pretrained on Electronic Health Records. We used Optuna (Akiba et al., 2019) for hyperparameter search and set our final configuration with a learning rate of $5e^{-5}$ using the AdamW (Loshchilov and Hutter, 2017) optimizer, a batch size of 64 and finetuned the models for 4 epochs. Ensembles of the same model used a different random seed when training each instance. We used 4 NVIDIA Tesla V100 with 32 GB of RAM with a training and inference time varying from 3 to 6.5 hours. A more detailed analysis of the training cost can be found in Appendix 6.

4.1.2 Prompting generative LLMs

We tested several Large Language Models (see Appendix F). We eventually chose Flan-T5-large[6] (Chung et al., 2022) for its ability to output answers that are easier to parse than the longer and more challenging answers that could be provided by Llama-2 (Touvron et al., 2023) or Mistral (Jiang et al., 2023). Flan-T5 has been pretrained on a mixture of 473 datasets covering 1,836 tasks. However, it has no biomedical or clinical pertaining. We rely on the HuggingFace framework for all experiments. We used the same computing setup as in the previous set of experiments. The codebase for all of our experiments is freely available.[7]

4.2 Evaluation

We evaluate our models using the following metrics. The F1 score of the *Entailment* class is measured on a control set of the gold test set which is the same as the NLI4CT 2023's test data. Faithfulness measures whether a model changes predictions when an 'entailing' statement is changed into a 'contradicting' statement. Consistency measures whether a model keeps its predictions when a statement is changed while preserving its relation to the premise. Both metrics are computed on a contrast set of the gold test set that has undergone perturbations (more details in Jullien et al. (2024)).

5 Results

5.1 Quantitative analysis

Under the username *math_agr*, our team ranked 27th for an F1 score of 0.57, 18th for Faithfulness of 0.64, and 25th for Consistency of 0.56. Tables 1–6 report the results of our experiments on the test set.

Single system	F1	Faithfulness	Consistency
Majority class	0.67	0.00	0.38
tf.idf (Jullien et al., 2024)	0.41	0.47	0.47
FZI-WIM	0.80	0.90	0.73
rezazzr	0.06	0.95	0.60
NYCU-NLP	0.78	0.92	0.81
a: NLI-RoBERTa	0.56	0.58	0.57
b: ClinicalBERT	0.00	**1.00**	**0.62**
c: Clinical-Longformer	**0.67**	0.00	0.38
Ensemble	s/h	s/h	s/h
(a+a+a)	0.57/0.57	0.58/0.54	0.57/0.56
(b+b+b)	0.56/0.63	0.37/0.16	0.47/0.43
(c+c+c)	**0.67**/0.64	0.00/0.09	0.38/0.40
d: (a+b+c)	0.55/0.57	0.45/0.40	0.52/0.52
(d) + Flan-T5-large	0.57 (h)	0.64 (h)	0.56 (h)

Table 1: F1 score, Faithfulness, and Consistency for single Masked Language Models then soft (s) and hard (h) ensembling. Ensembles such as (a+a+a) consist of 3 instances of the same model. Flan-T5-large is used in a 2S setting (see Tab. 2 below).

[2] https://huggingface.co/datasets/bigbio/sem_eval_2024_task_2

[3] https://huggingface.co/sentence-transformers/nli-roberta-base-v2

[4] https://huggingface.co/yikuan8/Clinical-Longformer

[5] https://huggingface.co/medicalai/ClinicalBERT

[6] https://huggingface.co/google/flan-t5-large

[7] https://github.com/MathildeAguiar/SemEval-2024-Task-2

Each model has different strengths and weaknesses across the three metrics in the MLM experiments. The single NLI-RoBERTa seems to be the most stable baseline despite its lack of pretraining on biomedical data. It has already been finetuned on general-domain NLI, and its sentence-level representation seems to boost its performance. The ensemble of 3 NLI-RoBERTa does not add enough diversity to improve its results. The single ClinicalBERT obtains an F1-score of 0.00: we observed that it always predicts the label *Contradiction*, which causes a precision and recall of 0.00. Faithfulness yields 1.00 because it is computed on instances of the contrast test set that are all labeled as *Contradiction*. The ensemble of 3 ClinicalBERT does not have this issue: some seeds led to better models. The single Clinical-Longformer obtains the best results in terms of F1-score but the worst on the other two metrics, especially on Faithfulness. It predicts almost exclusively *Entailment*, which leads to Faithfulness and Consistency complementary to ClinicalBERT's. The ensemble keeps the same issues. An ensemble (d) of the three single models could not improve the single NLI-RoBERTa. Adding Flan-T5's 2-shot predictions to the ensemble increased Faithfulness by 0.24 points but did not yield better F1. This did not improve either over Flan-T5 alone (see row 2S in Tab. 2).

Prompt	F1	Faithfulness	Consistency
ZS	0.56	0.57	0.55
1S	0.53	0.63	**0.57**
2S	**0.57**	0.64	0.56
1SCOT	0.39	0.70	0.53
2SCOT	0.43	0.69	0.51
1SCCOT	0.28	**0.85**	**0.57**
2SCCOT	0.24	0.81	0.56

Table 2: F1 score, Faithfulness, and Consistency for the LLM approach, using Flan-T5-large.

Prompting Flan-T5-large in few-shot mode performs as well as the fine-tuned NLI-RoBERTa. Increasing the number of demonstrations tends to improve the scores. This illustrates the usual trade-off between fine-tuning a smaller model or prompting a larger model without fine-tuning it. The Chain-Of-Thought method makes it more difficult to recognize *Entailment* relations and leads to lower F1. As seen above, this mechanically increases Faithfulness. Contrastive Chain-Of-Thought further reduces the number of predicted *Entailment* relations, with an associated increase in Faithfulness. All systems achieve similar Consistency.

The tf-idf baseline was provided by the task organizers. Some of our proposed systems scored below the baseline in some metrics. For instance, Clinical-Longformer obtained a much lower Faithfulness and Consistency, ClinicalBERT, CCOT, and 1SCOT prompts obtained lower F1 scores.

According to the leaderboard, the top scores were 0.80 for the F1 score, 0.95 for Faithfulness, and 0.81 for Consistency, achieved by 3 different teams. We do not have information regarding the approaches these teams chose at the time of writing. Using last year's results on the F1 score, the approach of Kanakarajan and Sankarasubbu (2023), using Flan-T5-xxl, achieved an F1 score of 0.83. Their approach differs from ours by not only prompting Flan-T5 but by finetuning it beforehand using single- and multiple-instruction templates. This approach leads to a boost in performance compared to our simpler approach. Takehana et al. (2023) also performed ensembling and voting of MLMs and achieved an F1 score of 0.66. They performed what we called 'hard voting,' using 10 models for their ensemble and performing data augmentation on the original task dataset. Their result is comparable to our approach using an ensemble of 3 ClinicalBERT or 3 Clinical-Longformer.

5.2 Error analysis

In this section, we analyze our models in more depth by breaking down their results according to gold labels, whether a comparison of CTRs is involved, the types of inference to perform, CTR sections, and examine the F1 score per intervention type. For simplicity, we focus our analysis only on the two best-performing systems of each approach.

Accuracy per gold label From the accuracy displayed in Tab. 3, we observe that our LLM methods, especially CCOT, handle the Contradiction examples better. This label is the most frequent in the test set (67% of instances labeled as Contradiction and 33% as Entailment). MLMs, in contrast, have similar accuracy across both labels.

Comparison versus Single The *Comparison* of 2 CTRs implies longer input sequences and possibly an increased complexity since the model needs to confront the elements of two separate documents. Surprisingly, as reported in Tab. 4, we observe that all models perform similarly for *Comparison* and

System	Entailment	Contradict.
3 NLI-RoBERTa	**55**	56
(d) + Flan-T5-large	**55**	48
2S	44	64
1SCCOT	20	**82**

Table 3: Accuracy (in %) per label: *Entailment* and *Contradiction* (Contradict.). Systems: ensemble of 3 NLI-RoBERTa; ensemble of all MLM baselines (d) + Flan-T5-large (2S); Flan-T5-large in 2-shot (2S) and 1-shot contrastive chain-of-thought (1SCCOT) settings.

Single. We can hypothesize that the models are able to find more clues with 2 documents instead of 1 and predict more accurate labels.

System	Single	Comparison
3 NLI-RoBERTa	56	56
(d) + Flan-T5-large	49	51
2S	59	56
1SCCOT	**61**	**61**

Table 4: Accuracy (in %) per CTR type: *Single* and *Comparison*. Systems: see Tab. 3.

CTR sections From the accuracy displayed in Tab. 5, we observe no performance distinction between the models for different sections.

System	AE	Int.	Elig.	Res.
3 NLI-RoBERTa	60	59	52	52
(d) + Flan-T5-large	43	46	55	57
2S	55	58	**61**	54
1SCCOT	**62**	**63**	58	**60**

Table 5: Accuracy (in %) per CTR section: *Adverse events* (AE), *Intervention* (Int.), *Eligibility* (Elig.), and *Results* (Res.). Systems: see Tab. 3.

Types of 'intervention' Tab. 6 results were obtained directly from the task organizers' evaluation script. Once again NLI-RoBERTa is stable across *Paraphrase* and *Definition* interventions and achieves the best performance. NLI-RoBERTa seems to be less sensitive to semantic change when it comes to paraphrasing. Its score for *Definition* shows that it can capture the relevant information better when more details are provided. Contrastive Chain-Of-Thought does not increase the model's resistance to semantic change (as shown by the results on *Paraphrase*), its ability to perform numerical inference (see results on Numerical paraphrase) or to focus on relevant information (see results on *Definition*). For the latter, the model might struggle to focus on relevant information because of the long length of the input prompts (see Tab. 11).

System	Def.	NP	Para.
3 NLI-RoBERTa	**0.57**	**0.51**	**0.56**
(d) + Flan-T5-large	0.39	0.46	0.54
2S	0.39	0.46	0.54
1SCCOT	0.31	0.26	0.25

Table 6: F1 score per intervention type: *Definition* (Def.), *Numerical Paraphrase* (NP), or *Paraphrase* (Para.) interventions. Systems: see Tab. 3.

6 Conclusion and future work

This paper describes the two systems proposed by the SEME team for the SemEval 2024 Task 2 NLI4CT. Our first approach is based on the finetuning and ensembling of Masked Language Models, using only the challenge's data. Our second approach consists of a pipeline to prompt Large Language Models, using prompt engineering techniques, such as Chain-Of-Thought and Contrastive Chain-of-Thought, in Zero-shot, 1-shot, and 2-shot manners. Our two best-reported results are 0.57 F1 score, 0.64 Faithfulness, and 0.56 Consistency, with prompting Flan-T5-large in a 2-shot manner, ranking 27th out of 32 submissions for F1, 18th for Faithfulness and 25th for Consistency. We obtain the same scores for the MLM system using an ensemble composed of a finetuned NLI-RoBERTa + Clinical-Longformer + ClinicalBERT + the predictions of Flan-T5-large, that is 0.57 for F1 score, 0.64 for Faithfulness, and 0.56 for Consistency.

Some future work could include the continuation of the Masked Language Models pretraining on unlabeled clinical trials, before performing a similar finetuning as presented in the paper. We could also apply this approach to medical Large Language Models like MEDITRON (Chen et al., 2023), by performing instruction-tuning using clinically oriented instructions and then prompting the resulting model on the task data. Another possible approach, similar to (Conceição et al., 2023), would be to incorporate domain ontologies (like UMLS) into the finetuning of Masked Language

Models to provide definitions and supplementary knowledge.

Ethical statement

The NLI4CT task uses clinical data extracted and processed from https://clinicaltrials.gov/. This resource is freely available, provided by the National Library of Medicine, and is an official U.S. Department of Health and Human Services website.

Carbon emissions

Another arguable ethical aspect of our approach is the carbon emissions generated by our models' training and inference. Our experiments used 4 Tesla V100 GPUs paired with 2 Intel Xeon Gold 6148 20 cores and 384 GB of RAM. Depending on the approach chosen, the running time can be up to 10 times longer. For instance, we observe an execution time of 3 hours for the training and inference of an ensemble of 3 ClinicalBERT models. For the inference of Flan-T5-large on a 2-shot Contrastive Chain-Of-Thought, we achieve up to 30 hours of running time to get the predictions for all instances of the test set. Globally, we can say that the MLM approach is computationally more efficient, with running times varying from 3 to 6.5 hours (for the ensemble of ClinicalBERT, NLI-RoBERTa, and Clinical-Longformer). For the LLM approach, we observe running times ranging from 10.5 hours (in Zero-shot) to 38 hours (in 1-shot Chain-Of-Thought).

We used Green Algorithms[8] (Lannelongue et al., 2021) to estimate carbon emissions, taking into consideration our aforementioned computational configuration. The MLM approach produces up to 831g of CO_2 with the 3 models ensembling approach. For the LLM approach, the emissions vary from 1.34 kg of CO_2 for zero, 1, and 2-shot experiments to 4.86kg for Contrastive Chain-Of-Thought experiments.

Considering the little gain in performance of LLMs compared to MLMs using our approach and the CO_2 overconsumption of the LLMs, it would be more reasonable to use the MLM approach in our case. The MLM approach also provides faster predictions, which can be much more convenient.

[8] http://calculator.green-algorithms.org/

Acknowledgements

This work benefited from the GPUs provided by Lab-IA, an institution member of Université Paris-Saclay. This work was also supported through the CNRS grant 80|PRIME.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. Meditron-70b: Scaling medical pretraining for large language models.

Yew Ken Chia, Guizhen Chen, Anh Tuan Luu, Soujanya Poria, and Lidong Bing. 2023. Contrastive chain-of-thought prompting. *ArXiv*, abs/2311.09277.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Sofia I. R. Conceição, Diana F. Sousa, Pedro Silvestre, and Francisco M Couto. 2023. lasigeBioTM at SemEval-2023 task 7: Improving natural language inference baseline systems with domain ontologies. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 10–15, Toronto, Canada. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud,

Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama AI research at SemEval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003, Toronto, Canada. Association for Computational Linguistics.

Loïc Lannelongue, Jason Grealey, and Michael Inouye. 2021. Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12).

Yikuan Li, Ramsey M Wehbe, Faraz S Ahmad, Hanyin Wang, and Yuan Luo. 2023. A comparative study of pretrained language models for long clinical text. *Journal of the American Medical Informatics Association*, 30(2):340–347.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Renze Lou, Kai Zhang, Jian Xie, Yuxuan Sun, Janice Ahn, Hanzi Xu, Yu su, and Wenpeng Yin. 2024. MUFFIN: Curating multi-faceted instructions for improving instruction following. In *The Twelfth International Conference on Learning Representations*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Bhavish Pahwa and Bhavika Pahwa. 2023. BpHigh at SemEval-2023 task 7: Can fine-tuned cross-encoders outperform GPT-3.5 in NLI tasks on clinical trial data? In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1936–1944, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Conner Takehana, Dylan Lim, Emirhan Kurtulus, Ramya Iyer, Ellie Tanimura, Pankhuri Aggarwal, Molly Cantillon, Alfred Yu, Sarosh Khan, and Nathan Chi. 2023. Stanford MLab at SemEval 2023 task 7: Neural methods for clinical trial report NLI. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1769–1775, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Guangyu Wang, Xiaohong Liu, Zhen Ying, Guoxing Yang, Zhiwei Chen, Zhiwen Liu, Min Zhang, Hongmei Yan, Yuxing Lu, Yuanxu Gao, Kanmin Xue, Xiaoying Li, and Ying Chen. 2023. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. *Nature Medicine*, 29.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

A Hyperparameters

Tab. 7 shows the final hyperparameters used for finetuning the Masked Language Model systems.

Hyperparameter	Value
Nb. epochs	4
Batch size	64
Learning rate	$5e-5$
Optimizer	AdamW

Table 7: Hyperparameters to finetune the MLM systems.

B Example of Natural Language Inference mechanism

Fig. 2 shows an example of the kinds of inference performed by the NLI system in order to predict the correct label.

C Prompts

C.1 Simple prompt

Fig. 3 displays an example Zero-shot prompt. For n-shot prompts, we insert n demonstrations before this prompt. Each demonstration is built from training data; in a demonstration, the Label part is replaced with 'Answer: Yes' or 'Answer: No' depending on whether the example's label is *Entailment* or *Contradiction*.

C.2 Chain-Of-Thought

Fig. 4 displays an example Chain-Of-Thought demonstration. Our initial demonstrations are modified to include the idea of Chain-Of-Thought as mentioned in Wei et al. (2022).

C.3 Contrastive Chain-Of-Thought

Fig. 5 displays an example of our Contrastive Chain-Of-Thought prompt. Our initial demonstrations are modified to include the idea of a Contrastive Chain-Of-Thought as mentioned in Chia et al. (2023).

D NLI4CT dataset statistics

Tab. 8 shows statistics regarding the original task's data, such as the number of CTRs, of statements, the average length of a statement or evidence, and the max length of an evidence or statement.

Metric	Value
Nb. CTRs (documents)	999
Nb. statements	2,400
Avg. length statement	19.5
Max. length statement	65
Avg. length evidence	10.7
Max. length evidence	197

Table 8: Statistics about the NLI4CT train and dev sets.

Subset	*Ent.*	*Cont.*
Train	850	850
Validation	100	100
Gold test set (whole)	1841	3659
Gold test set (control set)	250	250
Gold test set (contrast set)	1591	3409

Table 9: Statistics about the number of *Entailment* (Ent.) and *Contradiction* (Cont.) instances in NLI4CT dataset.

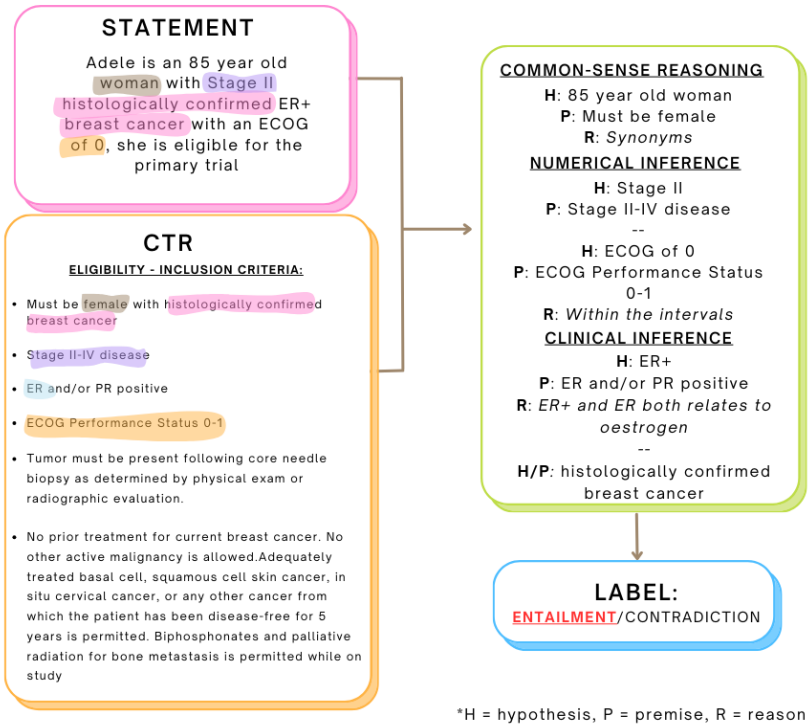


Figure 2: Example of an inference mechanism using a statement and the *Eligibility* section of a CTR.

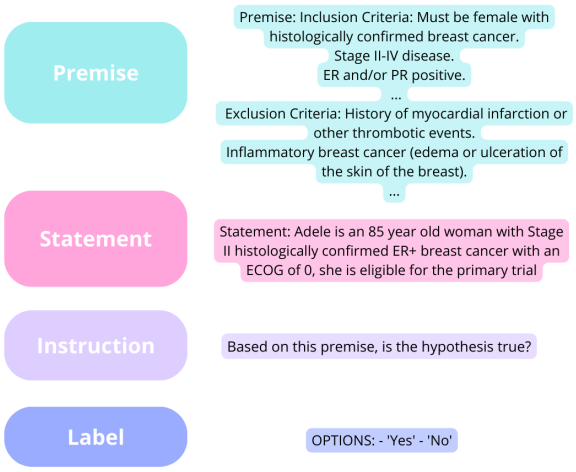


Figure 3: Example Zero-shot prompt.

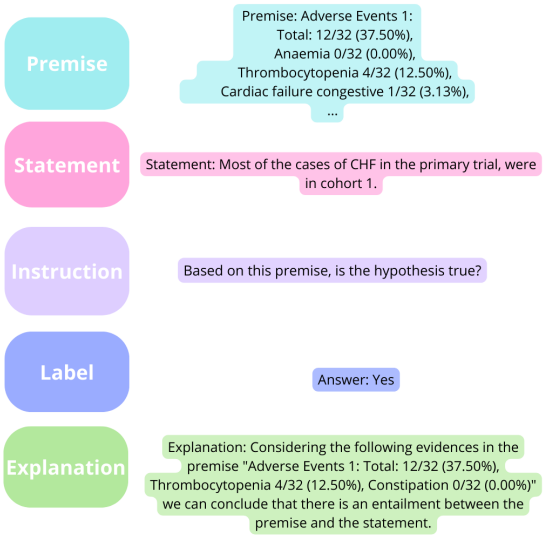


Figure 4: Example Chain-Of-Thought demonstration.

E Metrics on input sequences

E.1 MLM system input sequences

Tab. 10 displays the average, maximum, and minimum length of input sequences for the finetuning of MLMs.

E.2 LLM system input sequences

Tab. 11 displays the average, maximum, and minimum length of prompts used in Flan-T5.

F Prompt selection

Tab. 12 displays the templates tried in order to find the one that would perform the best. The last two prompts were tested using Llama-2 and Mistral. The last prompt uses the concept of 'persona prompting' (Zhang et al., 2018) where we assign the LLM a role.

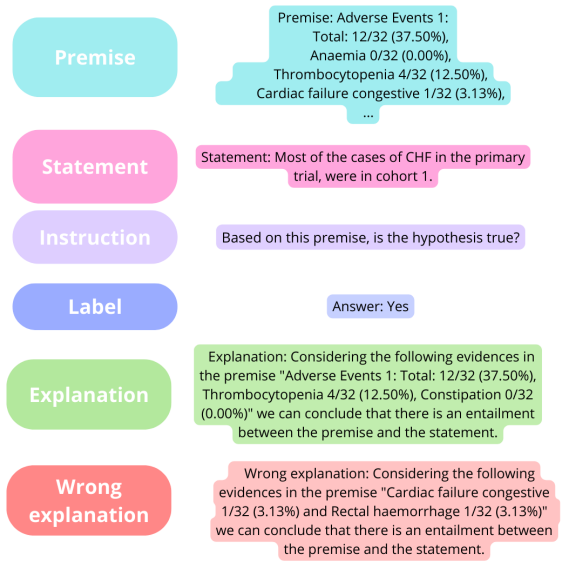


Figure 5: Example Contrastive Chain-Of-Thought demonstration.

Metric	Value
Mean nb. tokens	480
Min. nb. tokens	41
Max. nb. tokens	2799

Table 10: Average, minimum, and maximum number of tokens of an input sequence for the MLM approach.

Prompt	Mean	Min.	Max.
ZS	573	92	1367
1S	1650	835	3009
2S	3036	1397	6669
1S COT	2474	1300	6611
2S COT	3933	6354	2484
1S CCOT	2622	4285	1613
2S CCOT	4826	3153	8321

Table 11: Average, minimum, and maximum numbers of tokens of each kind of prompt for the LLM approach.

Id	Template
1	[Premise] [Statement] Does the premise entail the hypothesis? [Options]
2	[Premise] [Statement] Is the hypothesis entailed by the premise? [Options]
3	[Premise] [Statement] If this premise is true, what does that tell us about whether it entails the hypothesis? [Options]
4	From the following statement and premise, would you say there is a contradiction or an entailment between the statement and the premise? Just answer by saying 'contradiction' or 'entailment'. [Statement] [Premise]
5	Imagine you are a medical practitioner and you are reviewing clinical trials. You are given a statement and a premise. You should determine if there is an entailment or a contradiction between the premise and the statement. There is necessarily an entailment or a contradiction, no neutral case. From the following statement and premise, would you say there is a contradiction or an entailment between the statement and the premise? Just answer by saying 'contradiction' or 'entailment'. [Statement] [Premise]

Table 12: Other prompts tested on the LLM baselines.

MAINDZ at SemEval-2024 Task 5: CLUEDO - Choosing Legal oUtcome by Explaining Decision through Oversight

Irene Benedetto[1,2] **Alkis Koudounas**[1] **Lorenzo Vaiani**[1]

Eliana Pastor[1] **Luca Cagliero**[1] **Francesco Tarasconi**[2]

[1] Politecnico di Torino, {name.surname}@polito.it
[2] MAIZE, {name.surname}@maize.io

Abstract

Large language models (LLMs) have recently obtained strong performance on complex reasoning tasks. However, their capabilities in specialized domains like law remain relatively unexplored. We present CLUEDO, a system to tackle a novel legal reasoning task that involves determining if a provided answer correctly addresses a legal question derived from U.S. civil procedure cases. CLUEDO utilizes multiple collaborator models that are trained using multiple-choice prompting to choose the right label and generate explanations. These collaborators are overseen by a final "detective" model that identifies the most accurate answer in a zero-shot manner. Our approach achieves an F1 macro score of 0.74 on the development set and 0.76 on the test set, outperforming individual models. Unlike the powerful GPT-4, CLUEDO provides more stable predictions thanks to the ensemble approach. Our results showcase the promise of tailored frameworks to enhance legal reasoning capabilities in LLMs.

1 Introduction

Recent improvements in large language models are leading to a rethinking of legal practices, particularly in the United States (Frankenreiter and Nyarko, 2022; Hoffman and Arbel, 2023; Glaze et al., 2021). This can potentially transform time-consuming tasks such as brief writing and corporate compliance (Guha et al., 2023; Benedetto et al., 2023a). This could also contribute to alleviating the access-to-justice crisis (Corporation, 2017; Tito, 2017). The unique properties of LLMs, including their ability to learn from limited labeled data and proficiency in complex reasoning tasks, make them appealing for legal applications (Zheng et al., 2021; Guha et al., 2023; Benedetto et al., 2023b, 2024).

However, enthusiasm is tempered by concerns about the risks associated with LLMs, such as generating offensive, misleading, or factually incorrect content (Engstrom and Gelbach, 2020; Bender et al., 2021). These issues could have significant consequences, particularly affecting marginalized or under-resourced populations (Surden, 2020; Volokh, 2023; Koudounas et al., 2023, 2024).

To address safety implications, there is a pressing need to evolve and enhance legal reasoning capabilities in LLMs. Despite this urgency, practitioners face challenges in assessing LLMs' legal reasoning capabilities, as existing legal benchmarks are limited and often fail to capture the diverse aspects of legal tasks (Guha et al., 2023).

In this direction, the organizers of SemEval-2024 Task 5 introduce a novel Natural Language Processing (NLP) task and dataset derived from the U.S. civil procedure domain (Bongard et al., 2022). Each dataset instance comprises a case introduction, a specific question, and a potential solution argument, along with an in-depth analysis justifying the argument's applicability to the case. When provided with a topic introduction, a question, and a potential answer, the objective of the proposed task is to determine whether the given answer is accurate or not.

To tackle this task, we initially transform the dataset into a multiple-choice question answering problem using the multiple-choice prompting (MCP) approach (Robinson et al., 2023). We experimented with various open-source language models on this modified dataset, including Flan T5 XXL (Wei et al., 2021; Chung et al., 2022), LLama 7B and 13B (Touvron et al., 2023), Zephyr 7B (Touvron et al., 2023), and Mistral 7B (Jiang et al., 2023). Specifically, we trained these models to solve legal problems while also providing an explanation for the predicted outcome, leveraging the analysis provided. We thus introduce the *CLUEDO* approach, which stands for "**C**hoosing **L**egal o**U**tcome by **E**xplaining **D**ecisions through **O**versight". This framework utilizes multiple collaborative models to synthesize the final outcome based on each model's predictions. Each individual

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 997–1005

model is trained to predict the label of the correct candidate answer and generate an explanation accordingly. The final *"detective"* model operates in a zero-shot manner, relying upon the outputs of the collaborators. The model processes the answers and the explanations of all collaborators and deduces the ultimate answer.

The results on the challenge dataset demonstrate that our proposed methodology surpasses the performance of single models trained with standard fine-tuning. Furthermore, our approach achieved the second-place position in the public competition, achieving a final test F1 macro score of 0.77[1].

Research Questions. We investigate the following research questions (RQs):

- **RQ1.** Is the multiple-choice setting more effective than the single-choice one?
- **RQ2.** Does including the analysis in the training and generation process improve performance?
- **RQ3.** Is our detective model CLUEDO more effective than individual collaborators in a zero-shot setting? Are CLUEDO results more stable?

2 Related Work

In the legal domain, the advent of Legal LLMs has reshaped how legal professionals approach case analysis, decision-making, and document generation processes (Lai et al., 2023). LLMs possess logical reasoning capabilities that enable legal professionals to comprehend case processes, aid judges in decision-making, swiftly identify similar cases through language comprehension, analyze and condense essential case details, and utilize automated content generation to draft repetitive legal documents (Guha et al., 2023). Researchers have recently started exploring whether large language models have the capability to carry out legal reasoning. Unlike BERT-based models, LLMs are evaluated on their ability to learn tasks in-context, primarily through prompting (Liu et al., 2022). Studies have explored the role of prompt-engineering for Legal Judgment Prediction (Jiang and Yang, 2023), statutory reasoning (Blair-Stanek et al., 2023) legal exams (Yu et al., 2023). Several case studies (Nay et al., 2023; Drápal et al., 2023; Savelka, 2023; Savelka et al., 2023; Westermann et al., 2023) highlight the potential and the limitations of GPT models in real use cases. However, to the best of our knowledge, limited effort has been devoted to analyzing the effectiveness of smaller and open-source language models (e.g., Llama 2 (Touvron et al., 2023)) in this domain (Guha et al., 2023), and how they can effectively be employed in conjunction with closed-source foundational models, such as GPT-4 (OpenAI et al., 2023).

3 Dataset and Task Description

Bongard et al. (2022) present a new dataset from the U.S. civil procedure domain. This dataset is derived from a book intended for law students, suggesting its complexity and suitability for benchmarking modern legal language models. Each instance of the dataset consists of:

- *General introduction to the case*: an overview of the case to set the context.
- *Particular question*: a specific legal question related to the case is presented.
- *Possible solution argument*: a potential answer associated with the question is provided.
- *Annotated label*: it defines if the possible solution is correct (1) or not (0).
- *Detailed analysis*: Accompanying each solution argument is a thorough analysis explaining why the argument applies to the case in question.

The task is structured as a binary classification task where the goal is to predict the correctness of the answer provided, i.e., the label provided together with the textual information. The analysis and the labels are not available during test time.

4 System Overview

This section provides a comprehensive overview of the proposed methodology. Firstly, we outline the approach to the multiple-choice question-answering problem and how we adapt it to our scenario. Secondly, we introduce the *CLUEDO* framework, along with details about the competitors incorporated into our study.

[1]Code available at https://github.com/irenebenedetto/PoliToHFI-SemEval2024-Task5

Table 1: **Zero-shot** models on dev set. The best performance (in terms of F1 macro) for each model family is in bold. The multiple-choice approach leads to higher performance in five out of six cases.

Model	Classification task	Prec	Rec	F1	Acc
Flan T5 XXL	Multiple choice	0.60	0.67	**0.59**	0.64
Flan T5 XXL	Single choice	0.54	0.53	0.32	0.32
GPT-4	Multiple choice	0.66	0.73	**0.66**	0.57
GPT-4	Single choice	0.40	0.50	0.44	0.80
Llama 2 13B	Multiple choice	0.64	0.58	**0.59**	0.79
Llama 2 13B	Single choice	0.55	0.58	0.54	0.61
Llama 2 7B	Multiple choice	0.51	0.51	0.51	0.74
Llama 2 7B	Single choice	0.53	0.52	**0.52**	0.73
Mistral v0.1 7B	Multiple choice	0.55	0.59	**0.54**	0.61
Mistral v0.1 7B	Single choice	0.55	0.58	0.52	0.57
Zephyr beta 7B	Multiple choice	0.54	0.56	**0.50**	0.69
Zephyr beta 7B	Single choice	0.40	0.50	0.44	0.80

Table 2: **Trained** models performance on dev set. All models are trained to generate both labels and analysis, following the multiple-choice setting.

Model	Prec	Rec	F1	Acc
Llama 2 7B	0.57	0.60	0.56	0.64
Mistral v0.1 7B	0.61	0.63	0.62	0.73
Zephyr beta 7B	0.62	0.65	0.63	0.73
Llama 2 13B	0.65	0.69	**0.66**	0.75

Multiple-choice. Following the intuition of Robinson et al. (2023), we convert the dataset into a multiple-choice question answering problem and adopt multiple choice prompting (MCP) (Robinson et al., 2023). In MCP, the language model is presented not only with the question but also with a set of candidate answers, akin to a multiple-choice test. Each answer is linked to a symbol such as *"A," "B,"* or *"C."* This approach enables the model to compare answer choices explicitly and diminishes computational expenses for a generation. In cases where there is only one candidate answer, the system automatically generates the alternative *"None of the above is true"*. These additional answers are not accounted in the test and validation metrics.

In our experiments, we evaluate whether the multi-choice approach is indeed more effective than a single-choice approach. In the single-choice setting, we prompt a single choice, and the model should directly predict whether it is correct.

CLUEDO. To tackle the task of the challenge, we introduce the *CLUEDO* framework, which stands for "Choosing Legal Outcome by Explaining Decisions through Oversight." In a nutshell, multiple collaborative models are trained to predict the correct label for a candidate answer that addresses the legal question. These models generate their analysis as part of their training. The final model, operating in a zero-shot manner, utilizes the responses and explanations from the set of collaborators to identify the most accurate final answer, considering their collective performance. More in detail, the *CLUEDO* system is structured as follows:

- *N collaborative models*: given the introduction, the legal question, and the candidate answers, these models are trained to predict the label of the candidate answer that correctly responds to the legal question and generate an explanation. We fix the number of collaborators equal to three. We select the collaborators based on their results on the dev set.

- *The final "detective" model*: this model is employed in a zero-shot manner. Based on the responses from the collaborators and their corresponding explanations, this model must identify the most accurate final answer, overseeing the collaborators' performance. The final model is also provided with the introduction, legal questions, and candidate answers.

Example of prompts for collaborative and detective models are reported in Table 3.

Competitors. To assess the strength of the proposed CLUEDO approach, we compare the results with a set of alternatives on the final test set: the best collaborator chosen based on the results achieved on the dev set (that we call *Best collaborator*), and the correction of collaborator models based on consensus (after named *Collaborators agreement*). The latter approach involves taking the predictions of the top-performing collaborator (on the dev set) and rectifying instances where both the second and third collaborators mutually confirm inaccuracies. We finally employ the zero-shot final model without any collaborators to test its generalization capabilities, namely *Zero-shot detective model*.

5 Experimental Setup

Models. We evaluated various open-source models, employing both zero-shot and fine-tuning methodologies. Our analysis covered Flan T5 XXL (Wei et al., 2021; Chung et al., 2022), LLama 7B (Touvron et al., 2023) and 13B, Zephyr 7B (Touvron et al., 2023), and Mistral 7B (Jiang et al., 2023), selected for their unique features and performance metrics. Furthermore, we integrated into our assessment GPT-4 (OpenAI et al., 2023) in a zero-shot context.

Training procedure. We employed a Supervised Fine-Tuning (SFT) approach, implementing precision enhancement with 8-bit quantization. The models were trained for three epochs utilizing Parameter-Efficient Fine-Tuning (PEFT) (Mangrulkar et al., 2022), with a batch size set at 4 and a learning rate of 5e-5. The sequences were processed with a context length of 4096, optimizing the model's ability to capture long-range dependencies in the data.

Hardware. We run the experiments on a machine equipped with Intel® Core™ i9-10980XE CPU, 1 × Nvidia® Tesla T4 GPU, 16 GB of RAM running Ubuntu 22.04 LTS.

6 Results

To illustrate the efficacy of the multiple-choice setting and model selection criteria, we conduct individual tests for each configuration and present the obtained results on the development set. The following paragraphs address the research questions previously presented.

RQ1: Impact of the multiple-choice setting. Table 1 shows the zero-shot models' performance on the development set. For each model family, the multiple-choice question-answering approach consistently outperforms the single-choice approach in terms of F1 Macro. There is variability in the performance of different models within the same family. In general, larger models tend to exhibit stronger generalization capabilities than smaller ones.

RQ2: Impact of analysis inclusion in model training. In Table 4, we highlight the impact of including the analysis in the models' training process. To examine outcomes across various model sizes and classification tasks, we fixed the model family (Llama 2 from Meta). In both the 7B and 13B models, including the analysis (✓) consistently leads to higher performance for multiple-choice tasks. In particular, including the analysis during training leads to more balanced precision and recall metrics, resulting in an overall improvement in the F1 Macro score. For both Llama 2 7B and Llama 2 13B, the F1 Macro scores in single-choice tasks do not show significant improvement with the inclusion of the analysis. This may indicate that these models are less sensitive to additional analysis in single-choice tasks.

Additionally, the training of Llama 2 13B with the analysis allows for an additional $+0.07$ F1 score compared to its zero-shot counterpart, while for the 7B models, the training deteriorates the performance.

RQ3: CLUEDO results. The selection of collaborative models is guided by the results obtained on the development set as shown in Table 2. All models are configured to generate both labels and analysis, following the multiple-choice setting. Among the models, Llama 2 13B stands out with the highest F1 Macro score, indicating robust performance across multiple evaluation metrics, followed by Mistral and Zephyr models. For the supervisor model, we choose GPT-4, the best performer in the zero-shot setting (see Table 1).

Results on the test set are summarized in Table 5. Applying corrections based on the consensus of the second and third collaborators (Mistral and Zephyr) slightly reduces the F1 Macro to 0.65 on both development and test sets. This suggests that the initial collaborator's predictions were already quite accurate. The zero-shot model without collaborators

Table 3: **Example of prompts** for collaborative models and our CLUEDO approach.

Approach	Example Prompt
Collaborative Models	<s>[INST] <<SYS>>Given the following explanation and the question, which of the candidate answers is correct? The correct answer is the one that is true according to the explanation. <</SYS>> <explanation>Although discovery usually extends to all evidence relevant to claims and defenses in the action, Rule 26(b)(1) expressly carves out one [...] </explanation> <question>4. Confidential chat. Shag, a budding rock star with no business experience, enters into a five-year exclusive contract with Fringe Records, after [...] </question> <candidate_answers> 1 - Shag will not have to answer any of the interrogatories, because all three were discussed in å confidence with Rivera in the course of his representation. 2 - Shag will have to answer the first interrogatory, but not the other two. 3 - Shag will have to answer all three interrogatories, because [...] 5 - None of the above is true. </candidate_answers> [/INST] <correct_answer>5 </correct_answer> <analysis>Let's start by eliminating A. It proceeds on the premise that all three items are subject to discovery, because all [...] </analysis>
CLUEDO	You are a legal supervisor tasked with resolving legal queries. You are working alongside three artificial intelligence models, named m1, m2, and m3. Given an introductory context, a question, and a set of candidate answers, these three models must choose the correct answer and provide justification for their choice. Your responsibility is to assess the models' responses and determine whether they are correct or not. To do so, you must read the context (enclosed within the tags <context></context>), the question (within <question></question>tags), and the candidate answers (within <candidate_answers></candidate_answers>tags), and identify the correct answer among them (using the <supervisor_answer>tag). Additionally, you must provide reasoning for your choice (using the <supervisor_explanation>tag). While collaborating with the models and considering their advice, the ultimate decision rests with you. For each response, use the following format: <supervisor_answer>SUPERVISOR ANSWER</supervisor_answer> <supervisor_explanation>SUPERVISOR ANSWER</supervisor_explanation> <context>Although discovery usually extends to all evidence relevant to claims and defenses in the action, Rule 26(b)(1) expressly carves out one [...] </context> <question>4. Confidential chat. Shag, a budding rock star with no business experience, enters into a five-year exclusive contract with Fringe Records, after [...] </question> <candidate_answers> 1 - Shag will not have to answer any of the interrogatories, because all three were discussed in å confidence with Rivera in the course of his representation. 2 - Shag will have to answer the first interrogatory, but not the other two. 3 - Shag will have to answer all three interrogatories, because [...] 5 - None of the above is true. </candidate_answers> <m1_answer>1</m1_answer> <m1_explanation>[...] </m1_explanation> <m2_answer>1</m2_answer> <m2_explanation>[...] </m2_explanation> <m3_answer>2</m3_answer> <m3_explanation>[...] </m3_explanation> <supervisor_answer>

Table 4: **Trained** models on dev set. The best results (in terms of F1 Macro) are in bold. The generation of the analysis leads to higher performance for both 7B and 13B models.

Model	Classification task	Analysis included	Prec	Rec	F1	Acc
Llama 2 7B	Multiple choice	x	0.49	0.48	0.47	0.56
Llama 2 7B	Multiple choice	✓	0.57	0.60	**0.56**	0.64
Llama 2 7B	Single choice	x	0.40	0.50	0.44	0.80
Llama 2 7B	Single choice	✓	0.40	0.50	0.44	0.80
Llama 2 13B	Single choice	x	0.55	0.58	0.52	0.57
Llama 2 13B	Multiple choice	✓	0.65	0.69	**0.66**	0.75

Table 5: **Final Results** on dev and test sets: the best collaborator, collaborative agreements, and collaborators within CLUEDO are trained to generate the analysis along with the labels and adopt the MCP approach.

	Dev		Test	
Method	**F1**	**Acc**	**F1**	**Acc**
Best collaborator	0.66 (± 0.001)	0.75 (± 0.001)	0.69 (± 0.001)	0.75 (± 0.001)
Collaborators agreement	0.65 (± 0.001)	0.75 (± 0.001)	0.65 (± 0.001)	0.75 (± 0.001)
Zero-shot detective model	0.63 (± 0.038)	0.71 (± 0.024)	**0.77** (± 0.022)	0.83 (± 0.016)
CLUEDO	**0.74** (± 0.017)	0.78 (± 0.017)	**0.77** (± 0.017)	0.82 (± 0.013)

(GPT-4) performs well on the development set with an F1 score of 0.63. However, it surpasses all other methods on the test set with a notable F1 Macro of 0.77, showcasing its robust generalization capabilities. The CLUEDO model outperforms other methods with the highest F1 Macro on the development set (0.74) while achieving the second-highest score on test data. To assess the stability of predictions, we experimented five times on the validation set and test set and measured the performance of the models. Even with a greedy decoding strategy, small discrepancies regarding floating point operations lead to divergent generations, especially for larger models (Gawlikowski et al., 2021). It is known that this issue primarily concerns GPT-4[2]. Therefore, even though the temperature is set to 0 for all experiments, users have often reported significant variations in the output.

Although the predictions of trained models remained consistent, notable differences were observed in GPT-4 predictions, particularly when used without collaborators (the temperature is set to zero with no sampling). The results are presented in Table 5. With the proposed CLUEDO approach, the standard deviation is reduced by half. Additionally, the error estimate on the development set aligns with the one obtained on the test set. In conclusion, even though CLUEDO may not outperform others on test data, it ensures higher stability in predictions.

7 Conclusion

This paper presents a novel solution to the SemEval 2024 - Legal Reasoning Task, which introduced a challenge for evaluating contemporary legal language models. We transform the original dataset into a multiple-choice question-answering problem using the multiple-choice prompting approach and propose an original system, namely *CLUEDO*, that utilizes multiple collaborative LLMs and employs a final *"detective"* model to predict the outcome. Results show that our framework outperforms individual models in the public competition while returning more stable predictions, securing second place in the public competition.

[2]Here some discussion of the OpenAI community on models variability: https://community.openai.com/t/why-the-api-output-is-inconsistent-even-after-the-temperature-is-set-to-0/329541, https://community.openai.com/t/run-same-query-many-times-different-results/140588

References

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the

dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Irene Benedetto, Luca Cagliero, Francesco Tarasconi, Giuseppe Giacalone, and Claudia Bernini. 2023a. *Benchmarking Abstractive Models for Italian Legal News Summarization*.

Irene Benedetto, Alkis Koudounas, Lorenzo Vaiani, Eliana Pastor, Elena Baralis, Luca Cagliero, and Francesco Tarasconi. 2023b. PoliToHFI at SemEval-2023 task 6: Leveraging entity-aware and hierarchical transformers for legal entity recognition and court judgment prediction. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1401–1411, Toronto, Canada. Association for Computational Linguistics.

Irene Benedetto, Alkis Koudounas, Lorenzo Vaiani, Eliana Pastor, Luca Cagliero, Francesco Tarasconi, and Elena Baralis. 2024. Boosting court judgment prediction and explanation using legal entities. *Artificial Intelligence and Law*.

Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can gpt-3 perform statutory reasoning?

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Legal Services Corporation. 2017. The justice gap: Measuring the unmet civil legal needs of low-income americans.

Jakub Drápal, Hannes Westermann, and Jaromir Savelka. 2023. Using large language models to support thematic analysis in empirical legal studies.

David Freeman Engstrom and Jonah B Gelbach. 2020. Legal tech, civil procedure, and the future of adversarialism. *U. Pa. L. Rev.*, 169:1001.

Jens Frankenreiter and Julian Nyarko. 2022. Natural language processing in legal tech. *Legal Tech and the Future of Civil Justice (David Engstrom ed.) Forthcoming*.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna M. Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. 2021. A survey of uncertainty in deep neural networks. *CoRR*, abs/2107.03342.

Kurt Glaze, Daniel E Ho, Gerald K Ray, and Christine Tsang. 2021. Artificial intelligence for adjudication: The social security administration and ai governance.

Neel Guha, Julian Nyarko, Daniel E Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N Rockmore, et al. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *arXiv preprint arXiv:2308.11462*.

David A Hoffman and Yonathan A Arbel. 2023. Generative interpretation. *Available at SSRN*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Cong Jiang and Xiaolei Yang. 2023. Legal syllogism prompting: Teaching large language models for legal judgment prediction.

Alkis Koudounas, Eliana Pastor, Giuseppe Attanasio, Vittorio Mazzia, Manuel Giollo, Thomas Gueudre, Luca Cagliero, Luca de Alfaro, Elena Baralis, and Daniele Amberti. 2023. Exploring subgroup performance in end-to-end speech models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Alkis Koudounas, Eliana Pastor, Giuseppe Attanasio, Vittorio Mazzia, Manuel Giollo, Thomas Gueudre, Elisa Reale, Luca Cagliero, Sandro Cumani, Luca de Alfaro, Elena Baralis, and Daniele Amberti. 2024. Towards comprehensive subgroup performance analysis in speech models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S. Yu. 2023. Large language models in law: A survey.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

John J. Nay, David Karamardian, Sarah B. Lawsky, Wenting Tao, Meghana Bhat, Raghav Jain, Aaron Travis Lee, Jonathan H. Choi, and Jungo Kasai. 2023. Large language models as tax attorneys: A case study in legal capabilities emergence.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering.

Jaromir Savelka. 2023. Unlocking practical applications in legal domain: Evaluation of gpt for zero-shot semantic annotation of legal texts. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL 2023. ACM.

Jaromir Savelka, Kevin D. Ashley, Morgan A. Gray, Hannes Westermann, and Huihui Xu. 2023. Explaining legal concepts with augmented large language models (gpt-4).

Harry Surden. 2020. The ethics of artificial intelligence in law: Basic questions. *Forthcoming chapter in Oxford Handbook of Ethics of AI*, pages 19–29.

Joel Tito. 2017. How ai can improve access to justice.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan,

Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Eugene Volokh. 2023. Chatgpt coming to court, by way of self-represented litigants. *The Volokh Conspiracy*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Hannes Westermann, Jaromir Savelka, and Karim Benyekhlef. 2023. Llmediator: Gpt-4 assisted online dispute resolution.

Fangyi Yu, Lee Quartey, and Frank Schilder. 2023. Exploring the effectiveness of prompt engineering for legal reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13582–13596, Toronto, Canada. Association for Computational Linguistics.

Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 159–168.

Groningen Group E at SemEval-2024 Task 8: Detecting machine-generated texts through pre-trained language models augmented with explicit linguistic-stylistic features

Patrick Darwinkel, Sijbren van Vaals, Marieke van der Holt, Jarno van Houten
University of Groningen
Broerstraat 5, 9712 CP Groningen
{p.darwinkel, s.j.van.vaals, m.van.der.holt, j.t.van.houten}@student.rug.nl

Abstract

Our approach to detecting machine-generated text for the SemEval-2024 Task 8 combines a wide range of linguistic-stylistic features with pre-trained language models (PLM). Experiments using random forests and PLMs resulted in an augmented DistilBERT system for subtask A and B and an augmented Longformer for subtask C. These systems achieved accuracies of 0.63 and 0.77 for the mono- and multilingual tracks of subtask A, 0.64 for subtask B and a MAE of 26.07 for subtask C. Although lower than the task organizer's baselines, we demonstrate that linguistic-stylistic features are predictors for whether a text was authored by a model (and if so, which one).

1 Introduction

The SemEval-2024 Task 8 is aimed at the detection of machine-generated texts across different domains, languages, and generators. The challenge of distinguishing machine-generated from human written texts has become increasingly relevant with the rapid improvement and coinciding widespread usage of Large Language Models (LLMs) such as ChatGPT. Detection of machine-generated text can be important to uncover the purposeful spreading of misinformation on social media, or fraudulent articles and papers in the context of journalism and academics (Tang et al., 2023). To this end, the task organizers collect human data from Wikipedia, Reddit, Wikihow, PeerRead, and ArXiv abstracts in English, Chinese, Urdu, Russian, Indonesian, Arabic, and Bulgarian (Wang et al., 2023b). Consequently, Wang et al. (2023b) prompted 5 different generative LLMs to write the corresponding posts or abstracts based on the titles. The shared task consists of 3 subtasks. Subtask A is the task of classifying between human and machine-generated texts, subtask B pertains pointing out which LLM (or human) generated the text specifically, and lastly, subtask C is about determining the boundary where a text switches from human written to machine-generated.

The focus for our submission to the shared task is to investigate and compare the linguistic-stylistic characteristics of various LLMs, given that previous literature has shown that text produced by generative LLMs contain linguistic-stylistic anomalies (Tang et al., 2023). Additionally, we explore ways to combine features with the power of a pre-trained language model (PLM). Although a system inspired by linguistic-stylistic features may not achieve the greatest scores, it may perform well across domains and is highly interpretable. Additionally, the performance with linguistic-stylistic features may differ per LLM and per domain, for which they possibly yield interesting insights and contribute to scientific knowledge regarding what LLM-generated anomalies consist of.

Ultimately, our system yields passable results, coming in at 110 and 41 for subtask A mono- and multilingual respectively, and ranking at 46 for subtask B and 20 for subtask C. Contrary to expectations, the system appears to be relatively poor at generalizing between domains, but markedly better at dealing with multiple languages, as indicated by the increase in accuracy as well as ranking between the mono- and multilingual conditions in subtask A.

2 Background

To investigate which features contribute to the detection of machine-generated text, we collected 20 metrics from previous research which seem relevant. These features can be broadly divided into 6 categories, which will each be presented in this section.

2.1 Readability

Studies have shown that LLMs are capable of producing more readable text than human profession-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1006–1014

als when it comes to complex matters such as informed consent documentation (Decker et al., 2023). Pu and Demberg (2023) have also shown that, when it comes to producing summaries for layman or experts, ChatGPT tends to score very similar in both conditions, whereas human summarizers achieve lower scores for laymen, and higher in the expert condition. For this reason, we have selected 3 common, yet distinct readability formulas. The Flesch-Kincaid score for reading ease by taking the average sentence length, and the average number of sylables per word (Kincaid et al., 1975). Similarly, the Coleman-Liau index takes into account the average sentence and word lengths to compute a score (Coleman and Liau, 1975). Lastly, the Dale-Chall Readability Score is calculated using the average sentence length and the ratio of difficult words from a list to the total number of words (Chall and Dale, 1996).

2.2 Entity recognition

While previous research on using LLMs for Named Entity Recognition has shown promising results (Wang et al., 2023a), LLMs are still not as good as humans annotators. Therefore, we expect there may be a difference between human and machine-generated texts when it comes to entities. For this reason, we incorporate the ratio of entities to total words, as well as the ratio of unique to total number of entities as features.

2.3 Syntax

Syntax is concerned with the way words are put together to form proper sentences, often operationalized through dependency parsing where sentence constituents are labelled and linked to determine the syntactic structure of a sentence. Pu and Demberg (2023) used ChatGPT to transform texts from formal to informal and vice versa and found a clear difference between ChatGPT-generated and human-written text in the dependencies for both formal and informal sentences. Therefore, we include several metrics using dependency parsing. Firstly, we consider the average parse tree height (the length of the longest series of dependencies from the root constituent of a sentence). Additionally, we include the average number of noun phrases per sentence. Lastly we employ a measure of syntactic complexity, namely the Coh-Metrix SYNNP index (Graesser et al., 2004), which measures the mean number of modifiers per noun-phrase to compute complexity.

2.4 Semantics

Aside from syntactic features, previous research has also indicated differences on a semantic level. Firstly, machine-generated texts are less coherent than their human written counterparts (Tang et al., 2023). To make the concept of coherence measurable, we adopt the notion of lexical chains (Morris and Hirst, 1991), which refers to a series of related words that are linked by a common thread of meaning. The relevant features based on lexical chains are the total number, the average length and the span of lexical chains in a document. Furthermore, research has shown that ChatGPT produces less negative sentiment and offensive speech compared to human-authored texts (Tang et al., 2023), so we also include a score for controversy as a feature.

2.5 Text length statistics

As an extension of the readability metrics (see Section 2.1), which mostly combine different statistical features of texts to compute a score, we also take into account individual statistics of the document. Specifically, the average number of syllables per word, and the average sentence length.

2.6 Lexical Richness

LLMs work by selecting high-likelihood words to create coherent texts, making it likely for them to write using a lower diversity of words than humans. Previous research has shown that this is indeed the case (Guo et al., 2023). For this reason, we include a number of measures of lexical richness. Firstly, the type token ratio (TTR) and secondly, as an alternative to the TTR, which is sensitive to a steep drop-off in longer texts, the Measure of Lexical Diversity in Text (MLTD). To further study the lexical diversity put forth by LLMs, we consider the hapax richness of the document, which is the ratio of words in the text that occur only once. Lastly, we examine the ratio of function words to content words.

3 System overview

The backbone of our system is combining a feature-driven approach with the state-of-the-art in text classification, namely PLMs. To accomplish this, we tested three main system architectures for this shared task. Firstly, we augment DistilBERT with the feature set. Similarly, to achieve token-level classification, we perform the same for Longformer, and lastly, we use a Random Forest classifier with

DistilBERT embeddings in conjunction with the feature set. For comparison the features were also used separately and in combination with unigrams in a random forest classifier. The following sections will go into detail on each of these individually.

3.1 Augmented DistilBERT

For subtasks A and B, we augment `distilbert-base-cased` (Sanh et al., 2020) for the monolingual tasks and `distilbert-base-multilingual-cased` for the multilingual task with an additional layer for classification using features. The 20 features are run through a linear layer with ReLU activation. The output from this linear layer, which is equal in dimensions to DistilBERT's configured hidden size, is concatenated to the pooled output from DistilBERT's final hidden state. This results in a new tensor of 2*hidden size. This tensor is fed into another linear layer (2*hidden size, hidden size) with ReLU activation and dropout. The output from this is fed into the final classifier layer (hidden size, amount of labels).

3.2 Augmented Longformer

A challenge we ran into is the application of sentence-based features to token-level classification in subtask C. To address this problem, we use an augmented version of Longformer (Beltagy et al., 2020). The architecture of the Longformer is similar to the augmented DistilBERT, except that the output from the extra features is concatenated to the output state of each of the tokens separately. This essentially augments each token with contextual knowledge of the linguistic characteristics of the text that they occur in, enabling token-level classification.

3.3 DistilBERT-embedded Random Forest

Next to the augmented DistilBERT for subtasks A and B, we explore the use of a Random Forest classifier using `distilbert-base-cased` embeddings, instead of simpler one-hot encodings or TF-IDF embeddings, contatenated with our 20 linguistic-stylistic features. After tokenizing each text, we extract the DistilBERT embeddings for the first 512 sub-word tokens and average them using a concatenation of mean, max, sum and L2 (Euclidean norm) pooling. After retrieving the embeddings, we concatenate them with the feature vector composed of our 20 linguistic-stylistic features. We then use the concatenated embeddings with the features to fit a Random Forest classifier.

We experimented with different configurations which differed in the use of the layer (or hidden state) and pooling technique. We found the first hidden state layer (i.e., the layer after the input layer) using a concatenation of mean, max, sum and L2 pooling to produce the most satisfactory results. Similar to the augmented DistilBERT, we use `distilbert-base-multilingual-cased` in the multilingual track of subtask A.

4 Experimental setup

Much of the experimental setup is similar to the format dictated by shared task organizers (Wang et al., 2024). In particular, the provided train and dev sets were used as-is. However, some relevant aspects for our specific system will be presented in this section.

First and foremost, the features are extracted using a variety of external libraries, including SpaCy and fasttext, and subsequently put into JSON format. Furthermore, in the multilingual track, we recognize the language in question using Stanza and fasttext (how and why these models were used is explained in Appendix B.1), and apply language-specific feature extraction methods accordingly. Features that only work for English (e.g., Dale Chall) are not included in the multilingual track. These features were all assigned a value of -1. An overview of the features and how they were calculated can be found in Appendix B. To compare the features, we measured importance using Mean Decrease in Impurity (MDI), which computes the average change in homogeneity in Random Forest nodes for each feature. The systems will be evaluated using the official metrics of the task: accuracy for subtasks A an B and mean absolute error for subtask C.

5 Results

5.1 Development results

The results of our systems for subtasks A and B can be found in Table 1. For subtask A monolingual, augmented DistilBERT works best with an accuracy of 0.75, slightly better than the baseline of 0.74 from the organizers (Wang et al., 2024). Curiously, it performs worse than the default DistilBERT in the multilingual task. Where the default system had an accuracy of 0.71, the augmented version only had an accuracy of 0.67. Possible explanations are that not all features could be used multilingually

	Random Forest				DistilBERT	
	uni.	feat.	uni. + feat.	emb. + feat.	base	augm.
Subtask A monolingual						
Human	0.70	0.65	0.67	0.46		0.79
Machine	0.49	0.33	0.32	0.71		0.69
Accuracy	0.62	0.54	0.56	0.62	0.69	**0.75**
Subtask A multilingual						
Human	0.62	0.45	0.57	0.33		0.63
Machine	0.35	0.48	0.51	0.60		0.71
Accuracy	0.52	0.47	0.54	0.50	**0.71**	0.67
Subtask B						
Human	0.44	0.40	0.47	0.39		0.66
ChatGPT	0.74	0.57	0.73	0.59		0.77
Cohere	0.40	0.38	0.35	0.31		0.50
Davinci	0.58	0.11	0.40	0.24		0.29
Bloomz	0.93	0.89	0.89	0.83		0.95
Dolly	0.44	0.44	0.51	0.54		0.63
Accuracy	0.60	0.49	0.58	0.51	0.60	**0.64**

Table 1: F1-scores and accuracy for subtask A (monolingual and multilingual) and subtask B on the development sets for all our experiments: random forest with unigrams, features, unigrams + features and embeddings + features, and a base and augmented distilBERT model.

or that the quality of the multilingual features is worse than for the monolingual data, since not every language has the same quality parsing models available.

From the different implementations of the random forest model, the best accuracy on the monolingual task was achieved by both the unigram model and the model using DistilBERT embeddings and features with an accuracy of 0.62. There is a sizeable difference between the f1-scores of the 'human' and 'machine' labels. For the models using either unigrams, features or both, the 'human' label has an f1-score between 0.67 and 0.70, where the 'machine' label only scores between 0.32 and 0.49. Interestingly, adding the DistilBERT embeddings resulted in reversed scores. For this model the 'human' label only obtained an f1-score of 0.46 and the 'machine' label 0.71.

For the multilingual track the best random forest model was the one using unigrams and features which had an accuracy of 0.54. The model using only features performed worst with an accuracy of only 0.47. The reversal of the f1-scores of 'human' and 'machine' labels that occurred one the monolingual data is also present here but only when comparing the unigram model with the model using embeddings and features. The model using only features and unigrams and features both have similar scores for both labels.

Augmented DistilBERT also worked best for subtask B with an accuracy of 0.64. Of the random forest models, there was no model using features that outperformed the model using only unigrams. Of the different sources, davinci was the most difficult to predict with an f1-score of only 0.29 from augmented DistilBERT. The fact that the provided development set almost exclusively contains examples of Bloomz is clearly visible based on the comparatively high f1 scores for that class ranging between 0.83 and 0.95.

Since our features are document-based and subtask C is a token-level classification task, we only have have the results from our baseline Longformer and the augmented Longformer. The augmented system had a mean absolute error (MAE) of 5.29 on the development set. Much better than our non-augmented baseline system which had a MAE of 16.62, but still worse than the organiser's baseline of 3.53 (Wang et al., 2024).

5.1.1 Feature importance

Figure 1 shows the feature importances for subtasks A and B. Similar trends can be seen for both tasks, although there are some differences as well. Type-token ratio (and its extension, MTLD) and the amount and length of lexical chains are important for all tasks. The average number of syllables per word is very important for the monolingual track of subtask A, but not so much for subtask B. Unfortunately this feature could not be used for the multilingual track so a comparison is not possible. For further analysis, a correlation heatmap can be found in Appendix D.

5.2 Test results

The predictions of augmented DistilBERT were sent in for this task for both the monolingual and the multilingual track. For the monolingual track this resulted in an accuracy of 0.63, placing us at position 110 on the leaderboard. For the multilingual track of subtask A, the system performed better with an accuracy of 0.77, leading to position 41. Unfortunately both scores are lower than the organisers' baseline of 0.88 and 0.81 respectively.

For subtask B the same system was used as for subtask A: augmented DistilBERT. The accuracy on the test set was the same as on the development set, namely 0.64. This is lower than the organisers' baseline of 0.75 unfortunately. For this task it is interesting to look at a confusion matrix which is shown in Figure 2. It shows that there were a few

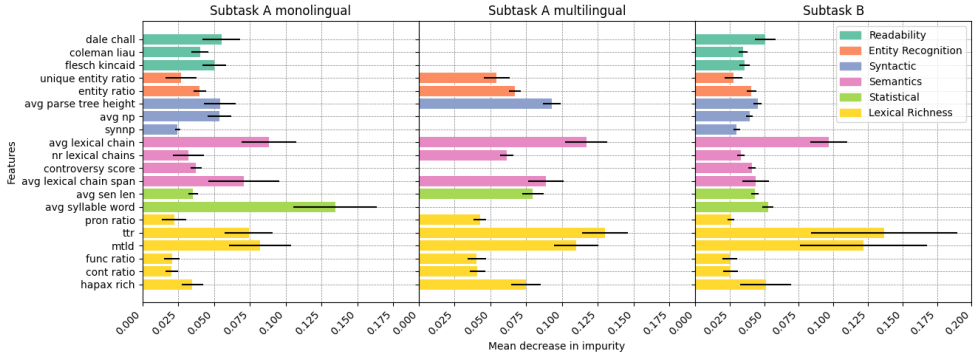


Figure 1: Feature importance (mean decrease in impurity from a random forest model using only our features) for subtasks A and B on the train set. The features are grouped by category.

generators that were problematic for the system. Cohere was almost never classified correctly and was mostly classified as ChatGPT. Dolly was often misclassified as davinci. Human written text was often classified as generated by either davinci or dolly, but almost never as the other three generators. For this subtask we obtained position 46 on the leaderboard.

We also evaluated the performance of our other main system for subtasks A and B, the random forest with embeddings and features, on the test set. This resulted in accuracies of 0.59 and 0.65 for subtask A (mono- and multilingual) and 0.39 for subtask B, also featured in Table 2. A possible explanation for the lower accuracies is that only the first 512 sub-word (BPE) tokens were taken into account. A lot of information gets lost during pooling and combing different strategies could not prevent a severe loss of information.

For subtask C the augmented Longformer achieved a MAE of 26.07, ranking us at position 20. Unfortunately this system also did not outperform the organiser's baseline of 21.54.

Subtask	Baseline	Augm. DistilBERT	Emb. RF
A mono	0.88	0.63	0.59
A multi	0.81	0.77	0.65
B	0.75	0.64	0.39

Table 2: Accuracy of the baseline, augmented DistilBERT and the embedded random forest on the test set for subtasks A and B.

6 Conclusion

Unfortunately none of our systems performed better than the baseline but our experiments did give some insight in how features can be used. Our final systems producing the most satisfactory results for the test set were: the augmented DistilBERT system for subtasks A and B, resulting in accuracies of 0.63 and 0.77 respectively, and the augmented Longformer for subtask C, obtaining a MAE of 26.07.

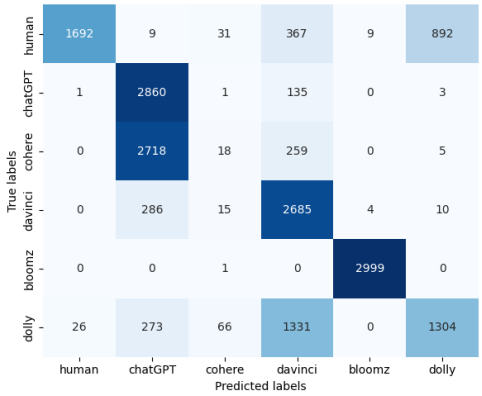


Figure 2: Confusion matrix from augmented DistilBERT for subtask B on the test set.

Contrary to the literature, it does not appear that the feature-based methods are better at generalizing. During our experiments we saw that when the unseen test data is from the same distribution (held from the training data), the feature-based approach performs far better in terms of accuracy (A mono: 0.88; A multi: 0.79; B: 0.71; for a full overview, see Appendix E). This is a clear indication that our proposed stylistic-linguistic features contain sufficient predictive power to distinguish human- from machine-written text. In particular, we find Type-token ratio, MTLD, the amount and length of lexical chains, and the average number of syllables per word to be noteworthy features for the given task. Future research into (different) features, especially for multilingual tasks could therefore be fruitful.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

J. S. Chall and E. Dale. 1996. Readability revisited : The new dale-chall readability formula., brookline books.

M. Coleman and T. L. Liau. 1975. A computer readability formula designed for machine scoring.

Hannah Decker, Karen Trang, Joel Ramirez, Alexis Colley, Logan Pierce, Melissa Coleman, Tasce Bongiovanni, Genevieve B. Melton, and Elizabeth Wick. 2023. Large Language ModelBased Chatbot vs Surgeon-Generated Informed Consent Documentation for Common Procedures. *JAMA Network Open*, 6(10):e2336997–e2336997.

Arthur C. Graesser, Danielle S. McNamara, Max M. Louwerse, and Zhiqiang Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, 36(2):193–202.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Dongqi Pu and Vera Demberg. 2023. ChatGPT vs human-authored text: Insights into controllable text summarization and sentence style transfer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 1–18, Toronto, Canada. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023a. Gpt-ner: Named entity recognition via large language models.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.

A Acknowledgements

This submission has been carried out as part of the 2023-2024 edition of the master course Shared Task Information Science (LIX026M05) at the University of Groningen, taught by Lukas Edman and Antonio Toral. We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Hábrók high performance computing cluster.

B Feature Extraction

In this section we describe how our features were calculated, what language models were used and how these models were selected. Version numbers of the specific libraries that were used can be found in appendix C.

B.1 Model selection

For the monolingual subtasks, we use the English `spacy_udpipe` model. This model is used to obtain POS-tags, noun chunks, syllable counts and parse trees. `en_core_news_sm`, also a spacy model, is used for named entity recognition.

For the multilingual task, the documents are first tagged with a language label by stanza's language identification model. This language label was then used to (try to) download the correct `spacy_udpipe` model and when this was not available, the correct stanza model. We did not use stanza models for all languages because they are quite large and slow.

For the NER tagger, the script first tries to download either `language_core_web_sm` or `language_core_news_sm`. When neither of these models is available, a universal model is used, namely `xx_ent_wiki_sm`.

For the fasttext embeddings the language of each document was first detected with ftlangdetect's `detect`. Because the spacy, stanza and fasttext models together take up quite some space, the data was processed per language. For this we chose the

stanza-identified language. For each language as it was identified by stanza, we chose the most occurring fasttext-detected language to download the correct fasttext model. It was not possible to use the stanza detected language for this as the language codes were not always identical and the languages supported by both are not the same.

B.2 Readability features

Flesch Kincaid

Note: this feature only works for English.
Calculated using `textstat`'s `flesch_reading_ease()`. The score is normalized by dividing the score by 100 (the maximum score) and subtracting this from 1 to invert the scale.

Coleman Liau

Note: this feature only works for English.
Calculated using `textstat`'s `coleman_liau_index()`. The score is normalized by dividing it by 30 (the maximum score).

Dale Chall

Note: this feature only works for English.
Calculated using `textstat`'s `dale_chall_readability_score()`. The score is normalized by dividing it by 20 (the maximum score).

B.3 Entity Recognition features

Entity Ratio

The number of entities divided by the number of words in a document.

Unique entity ratio

The number of unique entities divided by the total number of entities in a document.

B.4 Syntactic features

Average parse tree height

Average length of the longest series of dependencies from the root constituent of all sentences in the text.

Average number of noun phrases

Average number of noun chunks in a document.

SYNNP

Average number of tokens in a noun chunk in a document.

B.5 Semantic features

Lexical Chains

To create lexical chains only nouns were used. A chain is created by putting words together whose fasttext embedding have a cosine similarity (`sklearn`'s `cosine_similarity`) of more than 0.5. This feature is not used on its own, but to calculate the next three features.

Number of lexical chains

The sum of all lexical chains in a document. The score is normalized by dividing it by the number of words in a document.

Average lexical chain length

The average number of words in a lexical chain in the document. The score is normalized by dividing it by the number of words in a document.

Average lexical chain span length

The span is the number of words in the document between the first and last word of a lexical chain. Of this we take the average. The score is normalized by dividing it by the number of words in a document.

Controversy score

Note: this feature only works for English.
Calculated using `polarity_scores()` from `nltk`'s `SentimentIntensityAnalyzer()`.

B.6 Statistical features

Average number of syllables

Note: this feature only works for English.
Average number of syllables in a token.

Average sentence length

Average number of tokens per sentence (split by a period) in a document.

B.7 Lexical richness features

TTR

Calculated using `LexicalRichness().ttr` from `lexicalrichness`.

MTLD

Calculated using `LexicalRichness().mtld()` from `lexicalrichness`.

Hapax Richness

Calculated by getting `hapaxes()` from `nltk`'s `FreqDist()` and dividing it by the number of tokens in the document.

Content word ratio

The following POS-tags were used to select content words: "NOUN", "PROPN", "VERB", "ADJ", "ADV" and "NUM" from the universal POS tags and "CD", "JJ", JJR", "JJS", "POS", "PRP$", "RB", "RBR", "RBS", "WP$" and "WRB" from the Penn Treebank POS tags. The ratio is calculated by dividing the number of content words by the total number of words.

Function word ratio

Function words are all words that are not content words. The ratio is calculated by dividing the number of function words by the total number of words.

Pronoun ratio

Pronous are selected using the universal POS-tag "PRON" and the Penn Treebank POS-tags "PRP", "PRP$", "WP" and "WP$". The ratio is calculated by dividing the number of pronouns by the total number of words.

C Dependencies

The dependencies with version numbers that were used for both the feature extraction and the different system implementations.

- `fasttext`[1]
- `fasttext-langdetect`==1.0.5
- `lexicalrichness`==0.5.1
- `nltk`==3.8.1
- `numpy`==1.26.3
- `pandas`==2.1.4
- `scikit-learn`==1.3.2
- `spacy-udpipe`==1.0.0
- `spacy_syllables`==3.0.2
- `spacy_stanza`==1.0.4
- `stanza`==1.6.1
- `textstat`==0.7.3
- `datasets`==2.16.1
- `transformers`==4.36.2
- `accelerate`==0.25.0
- `evaluate`==0.4.1

[1]Recent Python and C++ versions require fasttext git:
fasttext @ git+https://github.com/facebookresearch/fastText@6c2204ba66776b700095ff73e3e599a908ffd9c3

Subtask A monolingual	
Human	0.89
Machine	0.87
Accuracy	0.88
Subtask A multilingual	
Human	0.78
Machine	0.80
Accuracy	0.79
Subtask B	
Human	0.76
ChatGPT	0.68
Cohere	0.68
Davinci	0.63
Bloomz	0.94
Dolly	0.54
Accuracy	0.71

Table 3: F1-scores and accuracy for subtask A (monolingual and multilingual) and subtask B from the random forest model using only features tested on a held out set of 20% of the training data.

D Feature multicolinearity

Figure 3 shows the multicolinearity of the features based on the training data set for the monolingual track of subtask A.

E Feature-based RF results on training data

Table 3 shows the results for subtasks A (monolingual and multilingual) and B when a random selection of 20% of the training set is held out and used as a test set for the random forest model using only features.

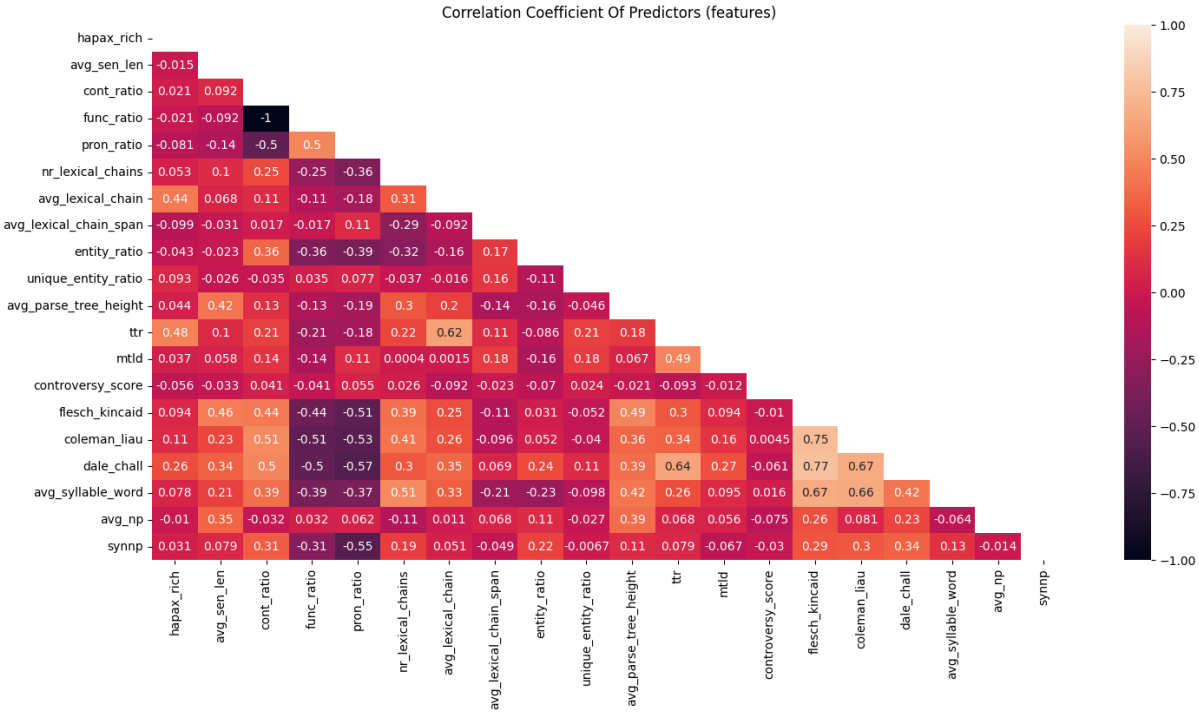


Figure 3: Heatmap showing the multicolinearity of the features based on the training data set of subtask A monolingual.

Magnum JUCSE at SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes

Adnan Khurshid and **Dipankar Das**
Department of Computer Science and Engineering
Jadavpur University, Kolkata, 700032, India

Abstract

This paper explores the detection of persuasion techniques within meme text, emphasizing logical fallacies and emotional appeals. Using a multilingual dataset structured as a directed acyclic graph, the study employs a node-level hierarchical classification with Support Vector Machines and pretrained sentence embeddings. Results demonstrate effective capture of nuanced persuasion techniques, providing fine-grained and general labels. The paper acknowledges dataset imbalance and assesses threshold impact on classification. The work contributes to understanding memes as conduits for persuasive communication, paving the way for future integration of image information for comprehensive analysis.

1 Introduction

In the realm of digital communication and social media, memes have emerged as a powerful and widely shared form of content, known for their ability to convey messages in a succinct and often humorous manner. While memes are commonly associated with entertainment, their potential as a tool for persuasive communication, particularly in the context of textual content, has become increasingly evident. This paper focuses on the nuanced task of detecting persuasion techniques within meme text in multiple languages like English, North Macedonian, Arabic and Bulgarian, exploring the ways in which textual elements contribute to the dissemination of persuasive messages.

The main strategy of the system is to train a binary classifier for each node in the hierarchy and predict labels in a top down fashion by seeing the confidence value of the prediction at any node. For each unique label in the hierarchy, a dataset is created from the original dataset which is then used to train the binary classifier for that label.

This task (Dimitrov et al., 2024) helped in understanding the intricacies of Hierarchical classification as well as sentence transformers. Our team participated in subtask 1 and ranked 21 out of 34 in English Language whereas 4 out of 20 in Bulgarian, 3 out of 20 in North Macedonian and 11 out of 17 in Arabic.

1.1 Objectives

The main objectives in this task include achieving the accuracy in classification of the internal nodes and minimising the number of classifiers and to look for a global classifier approach which takes the whole hierarchy into account at once. One more challenge due to having multiple levels of classes is handling the problem of inconsistency in predictions at different levels which means that the system may give negative prediction for some class at a level and then gives positive prediction for its children nodes. Since there are multiple output labels, Instances may belong to multiple classes that are not mutually exclusive or have overlapping characteristics due to the hierarchy being in the form of Directed Acyclic Graph. Distinguishing between such classes becomes complex

1.2 Contribution

The work done aims to create a model which performs the task of hierarchical multilabel classification of Persuasion techniques in memes with maximum accuracy. The model not only predicts the leaf nodes but also is able to predict corresponding internal nodes if the confidence in prediction is lower than some specified threshold at some node. Thus solving the class parent-child inconsistency problem stated earlier and providing a more robust and comprehensive classification of persuasion techniques in memes, enabling a deeper understanding of the hierarchical structure and allowing for enhanced decision-making based on varying levels of confidence in the predicted labels.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1015–1018

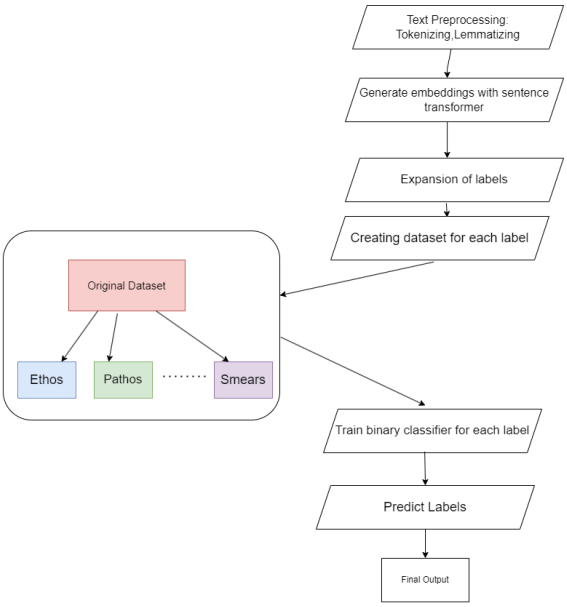


Figure 1: A diagram of the workflow.

2 Background

Our task involves the detection of persuasion techniques from memes. The data is provided in JSON format with string text and a list of string labels for each text. The labels are defined in a hierarchical manner in the form of a directed acyclic graph (DAG). A training set, a development or validation set, as well as a test set are available. The training and validation sets contain the labels while the test set only contains the text. In our dataset, 29 persuasion labels are defined.

The dataset is quite unbalanced, with some labels occurring many times, with others occurring much less. This can be partially attributed to the hierarchical nature of the data. Labels present neaerer to the root of the label DAG tend to appear much more frequently than the labels present nearer to the leaves of the DAG

3 System Overview

The step by step flow of the system is shown in the Figure [1] and explained in detail in the subsections which follow.

3.1 Overview

In this work, we use node level hierarchical classification. Our method consists of four major phases, data denoising, feature generation, node level classifier training and finally inference. Initially the data is cleaned and denoised, post this, features are generated for each of the sentences using a pre-trained sentence transformer. For classification, we consider a binary classifier at each node (Silla and Freitas, 2011) which predicts whether the example belongs to that node or not. We have employed the SVM (Support Vector Machine) as the classifier in our case.

Inference is done in a top-down fashion which the branch to be taken at each node is decided by the classifier at that node. This allows us to provide fine-grained as well as general labels. Fine grained labels are available toward the leaves of the tree and general labels are available towards the root. Based on the decision probabilities, we select the most suitable depth for the prediction results.

A final point worth mentioning is the identification of the threshold. Due to the imbalanced nature of the dataset , a threshold is determined using trial and error. The system works best with low threshold values for positive class because the training dataset for each unique label becomes highly skewed with negative examples.

3.2 Feature generation

Training a large language model from scratch on a corpus of strings requires very heavy computational resources, to which we did not have access. To circumvent this, we have utilized transfer learning, where the embeddings generated form a model on a general task is applied downstream effectively. This allows us to reuse previous work, if the task is sufficiently general, the pretrained model can produce very contextual and high quality embeddings.

For our current work, we have utilized the Sentence Transformer with Siamese BERT Embeddings as described in (Reimers and Gurevych, 2019). The authors of this paper have derived sentence embeddings in a contrastive manner utilizing similarity losses. Namely, they have utilized the triplet loss, which involves the creation of an anchor, a positive pair and a negative pair for embedding generation.

$$\mathcal{L}(A, P, N) = \max\left(d(A, P) - d(A, N) + \alpha, 0\right) \quad (1)$$

The goal of the Triplet Loss function is to minimize the distance between the anchor and the positive sample while simultaneously maximizing the distance between the anchor and the negative sample. A classification loss has also been utilized by the authors. Three labels have been considered, contradiction, neutral and entailment between

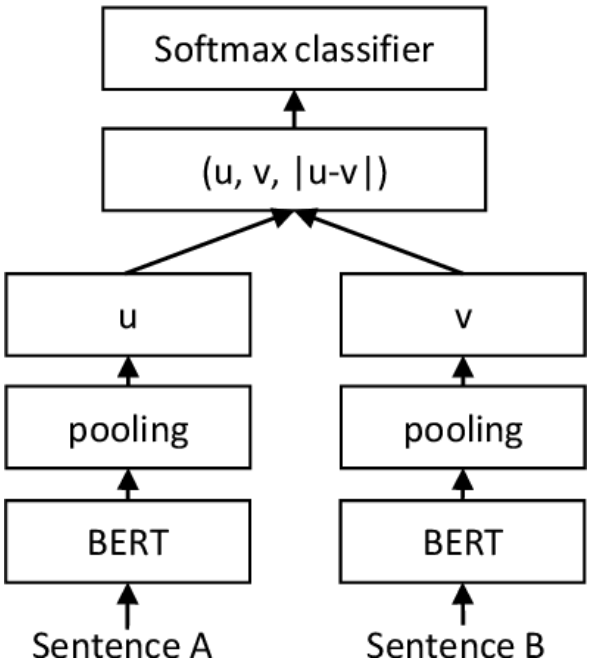


Figure 2: Architecture of the utilized sentence transformer.

pairs of sentences. The generated embeddings have length 768.

where:

- A represents the anchor sample,
- P represents the positive sample (same class as anchor),
- N represents the negative sample (different class from anchor),
- $d(A, P)$ denotes the distance between anchor and positive sample,
- $d(A, N)$ denotes the distance between anchor and negative sample,
- α is the margin, a hyperparameter that specifies the minimum difference between the distances.

3.3 Node Level Classifier

The data labels are represented in the form of a DAG. At each node, a SVM(support vector machine) is trained to predict whether the text instance belongs to that node or not. The node level classifiers are trained on the feature embeddings generated using the pretrained sentence transformer.

Support Vector Machines (SVMs) are powerful supervised learning models used for classification and regression tasks. The fundamental idea behind SVMs is to find the hyperplane that best separates the data points into different classes while maximizing the margin, which is the distance between the hyperplane and the nearest data points of each class.

SVMs can handle linearly separable as well as non-linearly separable data by employing the kernel trick, which maps the input data into a higher-dimensional space where it is easier to find a separating hyperplane. The optimization problem associated with SVM can be formulated as a convex optimization problem, typically solved using techniques such as quadratic programming.

4 Experimental Setup

4.1 Data Preprocessing

The input data is in the form of textual content for memes. There is a mix of capitalized, uncapitalized data as well as non-English words and gibberish. There is also the presence of arbitrary newlines in the dataset. To clean this data, firstly we have removed the unnecessary newlines in the data, replacing them with a single white-space , post this, we have removed all the punctuation. After this, we have lowercased the all the strings in the dataset, followed by stopword removal and lemmatization. This preprocessing improves the performance of the model as in general the dataset is very noisy and a model trained on it will not perform up to the mark.

4.2 Dataset Splitting

The original dataset is expanded by adding all the labels from root to leaf for a specific leaf label. So for example, if a row has label 'Slogans', then all the labels from root (Persusaion) to leaf (Slogans) are added, namely, Persuasion, Logos, Justification, Slogans and thus a dataset with expanded labels is formed. The dataset is then represented in One-Hot Encoding format for all the unique labels in the Hierarchy. So the dataset now contains 31 columns, 1 for the text, 1 for embeddings and 29 columns for the 29 labels in the hierarchy. So if a row has labels [Persuasion, Logos, Justification, Slogans] then the columns of these labels will have value 1 and others will have 0. Then a set of smaller datasets with the columns text,embeddings and the binary output for each label is created from the original dataset. These datasets are stored in a dictionary in key-value pairs where the key is the label and value

is a dataframe containing the dataset. Thus for the 29 unique labels in the hierarchy, 29 datasets are created.

4.3 Inference

The classification is done in two ways. First the text embedding is passed through all leaf node classifiers and the labels which give positive prediction with confidence greater than 0.7 are directly added to the output. Secondly, we then pass the embedding to a function which does the classification in a top down or depth first approach. We start from the root by pushing the children nodes of the node which has positive prediction confidence greater than the predefined threshold value to a stack. Then we pop from the stack and keep repeating until a leaf node is reached or the prediction confidence is very low at any particular node. The distinct labels from both these are then taken as the final output.

5 Results

We have provided the results of our method using some different thresholds. The result contains Hierarchical F1 Score, Hierarchical Precision as well as Hierarchical Recall. How the threshold is set is explained in the table [1]. A confusion matrix is shown for the prediction of leaf nodes in Figure [3] The test results for the languages Bulgarian and North Macedonian after final submission are also shown in tables [2] and [3]

Threshold	Hierarchical F1	Precision	Recall
For Depth = 0 : 0.3			
For Depth = 1 : 0.4			
For Depth $\geq$ 2 : 0.5	0.5624	0.6322	0.5065
All nodes : 0.24	0.6034	0.5465	0.6734

Table 1: Results for different threshold values at different depths of the hierarchy.

Threshold	Hierarchical F1	Precision	Recall
All nodes : 0.24	0.49986	0.47027	0.53342

Table 2: Final submission result on test data in Bulgarian Language.

Threshold	Hierarchical F1	Precision	Recall
All nodes : 0.24	0.48267	0.48568	0.47970

Table 3: Final submission result on test data in North Macedonian Language.

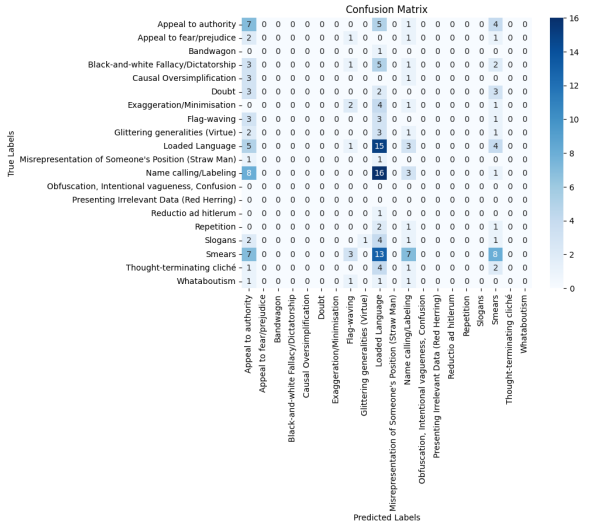


Figure 3: Confusion matrix for only leaf node predictions.

6 Conclusion and Future Work

The system gives satisfactory results on the validation dataset but more testing is required to measure the accuracy of the model. The accuracy of classifiers for some of the internal nodes is low because of a large variety of text sentences corresponding to the internal labels The leaf node classifiers generally have very high accuracy due to low number of example instances

This system only works with textual data, considering memes have rich image information as well, utilizing it in sync with the textual data to accurately predict persuasion techniques would be a natural continuation of this work.

References

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data mining and knowledge discovery*, 22:31–72.

Tübingen-CL at SemEval-2024 Task 1: Ensemble Learning for Semantic Relatedness Estimation

Leixin Zhang
University of Tübingen, Germany
leixin.zh@gmail.com

Çağrı Çöltekin
University of Tübingen, Germany
cagri.coeltekin@uni-tuebingen.de

Abstract

The paper introduces our system for SemEval-2024 Task 1, which aims to predict the relatedness of sentence pairs. Operating under the hypothesis that semantic relatedness is a broader concept that extends beyond mere similarity of sentences, our approach seeks to identify useful features for relatedness estimation. We employ an ensemble approach integrating various systems, including statistical textual features and outputs of deep learning models to predict relatedness scores. The findings suggest that semantic relatedness can be inferred from various sources and ensemble models outperform many individual systems in estimating semantic relatedness.

1 Introduction

Identifying semantic relatedness is a 'related' task to many well-studied tasks of semantic similarity. According to Abdalla et al. (2023), two sentences are considered similar if they are paraphrases or share a relation of entailment. Semantic relatedness, however, is a broader concept than semantic similarity. Two expressions are considered related if they share any semantic association. For instance, 'teacher' and 'student' are related because they frequently occur within the same context or domain. Similarly, 'tasty' and 'unpalatable' are related, as both terms are used to describe food, albeit with opposite meanings.

SemEval-2024 Task 1 (Ousidhoum et al., 2024b) is designed to estimate the relatedness of sentence pairs. The task is based on a multilingual dataset of 14 languages and offers supervised, unsupervised and cross-lingual tracks. Our team participated in two tracks, and a subset of available languages: Track A (supervised learning) for English, and Track B (unsupervised learning) for English, Spanish, and Hindi.

We posit that semantic relatedness can be inferred from a multitude of sources and therefore propose an ensemble approach that integrates outcomes from diverse systems to estimate semantic relatedness. Our study explores features from textual statistical analysis, general large language models, word embedding models, and models trained on semantic labeled datasets, question-answering pairs, or title-passage pairs in estimating semantic relatedness, and we conducted ensemble experiments with these features.

2 Related Work

SemEval in previous years has introduced tasks focusing on semantic textual similarity to evaluate the degree of similarity between sentence pairs (Agirre et al., 2012; Manandhar and Yuret, 2013; Agirre et al., 2014; Cer et al., 2017). There tasks provided datasets with human labeled similarity scores, which have been extensively utilized for training sentence embedding models and conducting semantic evaluations (Wieting et al., 2015; Cer et al., 2018; Reimers and Gurevych, 2020; Feng et al., 2022).

2.1 Sentence Embeddings

Word embedding models such as BERT (Devlin et al., 2019), GloVe (Pennington et al., 2014), RoBERTa (Liu et al., 2019), and Word2Vec (Mikolov et al., 2013) are frequently employed to assess the semantic distance between words. Sentence embeddings with a fixed length are often generated via mean/max pooling of word embeddings or employing `CLS` embedding in BERT. The semantic distances are commonly measured using the cosine similarity of embeddings of two expressions.

Siamese or triplet network architectures are frequently employed in sentence embedding training. For example, models such as Sentence-BERT (Reimers and Gurevych, 2019, 2020) utilize a dual-encoder architecture with shared weights for

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1019–1025

predicting sentence relationships (e.g., semantic contradiction, entailment, or neutral labeling) or for similarity score prediction using regression objectives, e.g., the difference between human annotated similarity score (sim) of two sentences and the cosine of two sentence embeddings (v and u), illustrated in Equation (1).

$$\mathcal{L} = |\cos(v, u) - \text{sim}| \tag{1}$$

In triplet neural networks, an anchor sentence (u) can be trained along with a positive sample (a sentence with a similar meaning) and a negative sample (a sentence with a dissimilar meaning), with contrastive loss. InfoNCE (Noise-Contrastive Estimation) can be utilized as the objective function. A larger number of negative samples can also be integrated into neural networks through the application of InfoNCE, as demonstrated in Equation (2). Here, v^+ denotes positive samples. The negative sample size is denoted as K, and the total sample size (including one positive sample) as $K + 1$. This approach is adopted by the Jina embedding model (Günther et al., 2023), which is used in our ensemble system.

$$\mathcal{L} = -\mathbb{E}\left[\log \frac{f(v^+, u)}{\sum_{i=1}^{K+1} f(v_i, u)}\right] \tag{2}$$

2.2 Ensemble Learning

In previous studies, ensemble learning presents several advantages. The ensemble approach can reduce the errors from individual models by amalgamating results from multiple sources or can make the system more robust. In our study, using multiple pre-trained models can also save a substantial amount of computation while making use of information from the large data during pre-training. Previous research has demonstrated that ensemble learning can achieve remarkable success (Huang et al., 2023; Osika et al., 2018).

In our study, we aim to integrate multiple deep learning models to assess semantic relatedness. When models are trained on diverse datasets with different architectures, they may produce varied predictions on semantic relatedness, and combining them may improve overall performance.

We use sentence embeddings mainly from the following models. Sentence-BERT (Reimers and Gurevych, 2019) is trained on datasets involving SNLI (a collection of 570,000 sentence pairs) and MultiNLI (comprising 430,000 sentence pairs). The Jina Embedding model (Günther et al., 2023) utilizes 385 million sentence pairs and 927,000 triplets (comprising positive and negative samples of semantic similarity) after a filtering process. The T5 model is trained on approximately 7 TB of text data derived from Common Crawl, serving various text-to-text purposes (Raffel et al., 2020; Ni et al., 2021).

3 Methodology

In this study, we hypothesize that semantic relatedness covers a broader spectrum than semantic similarity in theory. Consequently, the integration of various systems and features should achieve superior results compared to individual systems.

3.1 Supervised Learning

For the supervised track[1], we first evaluated subsystems in an unsupervised manner and selected those with a higher Spearman's correlation with human annotations for ensemble learning. The selected results were then further fine-tuned using the training data (5,500 English sentence pairs labeled with relatedness scores provided by the shared task, Ousidhoum et al., 2024a) to achieve closer alignment with human annotations.

In the following subsections, we present the features and systems utilized for ensemble learning. The features can be classified into three categories: textual statistical features (Section 3.1.1), word embedding models (Section 3.1.2), and sentence embedding models (Section 3.1.3).

3.1.1 Textual Statistical Features

Our analysis began with surface-level textual statistical features, including word overlap and the Levenshtein distance measurement at the character level. These scores were then normalized into ratios to estimate their correlation with human-annotated relatedness. Specifically, we considered the following features:

- Character Distance Ratio: normalization of Levenshtein distance. Levenshtein distance (represented as $Dist$ in Equation (3)) or edit distance is a string metric for measuring the

[1] In the supervised track, we only participated English subtask, in which relatively more training data was provided. For this reason, our analysis of supervised learning is specific to English.

Statistic Features	Spearman *r*
Char Distance Ratio	0.513
Word Overlap Ratio	0.593
Content Words Overlap Ratio	0.604

Table 1: Correlation between human-annotated relatedness scores with ratios of textual statistical features.

difference or distance between two sequences at the character level. The character ratio we use in this study is defined as:

$$\frac{\text{len}(\text{Sent}_1) + \text{len}(\text{Sent}_2) - \text{Dist}}{\text{len}(\text{Sent}_1) + \text{len}(\text{Sent}_2)} \quad (3)$$

- Word Overlap Ratio: the count of overlapped words over the total word count in sentence pairs, expressed as:

$$\text{Ratio} = \frac{|\text{Words(A)} \cap \text{Words(B)}|}{|\text{Words(A)} \cup \text{Words(B)}|} \quad (4)$$

- Content Word Overlap Ratio: the overlap ratio with content word considered only. Content words and functional words are distinguished by analyzing their part-of-speech (POS) using `SpaCy` python package.

We found that the overlap ratio computed solely on content words shows a better correlation with the human judgment of relatedness (Table 1). Furthermore, we tested the correlation of the word overlap ratio with the other two scores: Spearman's *r* with content word overlap ratio is 0.77, and Spearman's *r* with character distance ratio is 0.78. This suggests that the combination of two or more results may improve the relatedness estimation.

3.1.2 Word Embedding Models

In this subsection, we evaluate the performance of word embedding models' potential to estimate semantic relatedness. Sentence embeddings are represented as the mean of the word embeddings of all words in the sentence. We explored static word embeddings (GloVe and first layer BERT embeddings) and contextual word embeddings (the last layer of BERT embeddings) in relatedness estimation. The performance of the following variations is presented in Table 2:

- PCA transformation of embeddings. By using the PCA technique, we do not intend to reduce the dimension of the sentence embeddings, but transform sentence embeddings onto a new coordinate system such that the principal components capture the largest variation in the data. In practice, the maximum dimension that fits the dataset is adopted: *min* (*embedding_length, sample_size*).

- Content word embeddings: the average of word embeddings of content words only.

- Noun embeddings: the average of word embeddings for nouns only.

- Tree-Based word embeddings: the mean of embeddings of words that are at the top three levels of dependency trees,[2] namely the root (main predicate), direct dependents of the root, and dependents with the dependency distance of 2 from the root.

Our preliminary analysis offers the following insights for further ensemble learning:

1. Excluding functional words (using content words only) can enhance the effectiveness of GloVe embedding.

2. Focusing on words closer to the sentence's 'root' in terms of dependency distance did not yield better results.

3. Contextualized BERT embeddings do not necessarily outperform uncontextualized embeddings in semantic relatedness estimation.

4. PCA-transformed embeddings show improved correlation with human annotation of relatedness.[3]

3.1.3 Models for Sentence Representations

For supervised learning, we also incorporate sentence representations from pre-trained language models into our ensemble system. This includes models known for their strong performance in sentence similarity tasks, involving Sentence-BERT (`mpnet-base`, Reimers and Gurevych, 2019) and Jina Embedding (`jina-v1`, Günther et al., 2023), as well as the general large language model, T5

[2]We use SpaCy to parse sentences and select the root and dependents

[3]Despite the better performance of PCA-transformed embeddings in Spearman's correlation when word embedding models are tested individually, it was not beneficial in later supervised training. Ultimately, $\text{GloVe}_{\text{Content}}$ word embedding was utilized in supervised and unsupervised ensemble learning for English.

Model	Spearman *r*
GloVe	0.460
$\text{GloVe}_{\text{PCA}}$	0.533
$\text{GloVe}_{\text{Content-words}}$	**0.554**
$\text{GloVe}_{\text{Tree-Based}}$	0.249
$\text{GloVe}_{\text{Noun}}$	0.430
$\text{BERT}_{\text{LastLayer}}$	0.399
$\text{BERT}_{\text{LastLayer/PCA}}$	0.446
$\text{BERT}_{\text{FirstLayer}}$	0.570
$\text{BERT}_{\text{FirstLayer/PCA}}$	**0.593**

Table 2: Spearman's correlation between human-annotated relatedness scores with the cosine similarity of average embeddings of all words, content words, all nouns or tree-based word selections within a sentence. PCA-transformed average embeddings of all words in a sentence are also presented.

encoder (Raffel and Chen, 2023; Ni et al., 2021). Among all models tested in this study for English (refer to Table 3), T5 demonstrates the highest performance, achieving a Spearman's correlation of approximately 0.82 with human annotation.

3.1.4 Ensemble Learning

We explored two approaches for ensemble learning. The first approach operated directly on sentence representations from multiple models. This included concatenating sentence embeddings from various models and applying transformation (e.g., PCA transformation) in the embedding space to achieve a better correlation with human judgment. Our analysis indicates that while concatenation and transformation operations can slightly improve Spearman's correlation, they are not as effective as incorporating more statistical features into supervised fine-tuning.

In the final system, we directly used the cosine similarity values from sentence embedding and word average embeddings as features (from models `mpnet-base`, `jina embedding`, `T5-base` and mean of content word embeddings from GloVe), along with textual statistic features (content word overlap ratio and character distance ratio) to estimate the relatedness of sentence pairs. These features are fed into Support Vector Machine (SVM) regression models (with RBF kernel) to predict human annotated relatedness.

3.2 Unsupervised Ensemble

In the unsupervised track, without utilizing labeled datasets for sentence similarity or relatedness and without employing models pre-trained on labeled datasets, we aim to evaluate whether models trained on other types of datasets intended for different purposes could generate representations suitable for estimating semantic relatedness.

In addition, we investigated whether integrating additional features, such as the cosine distance of average word embeddings and word overlap ratios, could enhance performance. We calculated the arithmetic mean of the cosine distances and ratios from textual statistics as the relatedness prediction of sentence pairs. Various feature combinations are tested with the provided validation dataset.

For the unsupervised task of English, we utilized two models to generate sentence representations: a model designed for semantic search (`multi-qa-MiniLM-L6-cos-v1`, Reimers and Gurevych, 2019), trained on 215 million question-answer pairs; and e5 (`e5-base-unsupervised`, Wang et al., 2022),[4] trained on question-answer pairs, post-comment pairs, and title-passage pairs. These models were further refined with an unsupervised transformation (PCA). Additionally, we incorporated two other features: PCA-transformed GloVe embeddings (average of content word embeddings within a sentence) and content word overlap ratios into the unsupervised ensemble system.

For the unsupervised tasks in Spanish and Hindi, we used a similar method for predicting relatedness, combining features involving the cosine distance of `multi-qa-MiniLM` model representations, word embedding model and word overlap ratios. For word embeddings, we employed multilingual BERT (`bert-base-multilingual-uncased`), utilizing both the first-layer (uncontextualized) and last-layer (contextualized) embeddings for relatedness estimation.

4 Results and Analysis

The shared task evaluates the participating systems based on Spearman's correlation (r) between the human-annotated scores, which ranges from 0 to 1. In Table 3, we compare the correlation scores for our systems and other popular models on the official test set.

[4]The e5 monolingual model is exclusively used for English, not for the other two languages: Spanish and Hindi

Models	English	Spanish	Hindi
Lexical Overlap	0.741	0.661	0.587
$\text{mBERT}_{\text{\_Ave}}$	0.640	0.655	0.566
mpnet-base[5]	0.809	0.590	**0.746**
T5 (base)	0.825	-	-
LaBSE[6]	0.818	0.651	0.709
multi-qa-Mini	0.793	0.638	0.466
$\text{Ensemble}_{\text{Sup}}$	**0.850**	-	-
$\text{Ensemble}_{\text{Unsup}}$	**0.837**	**0.705**	0.649

Table 3: Spearman correlation between human-annotated relatedness scores and system predicted scores on the test dataset.

Results presented in Table 3 suggest that the ensemble approach generally outperforms single models. Specifically, the ensemble system trained with true labels, for the supervised English task, achieved the best result among all listed systems, with an improvement in Spearman's correlation of 0.025 compared to the T5 base model.

The ensemble approach for English and Spanish unsupervised tasks also achieved relatively high scores, despite the absence of similarity or relatedness scores in learning. It suggests that semantic relatedness can be estimated without necessarily relying on human-annotated scores of semantic similarity or semantic relatedness. Other sources like question-answering pairs or statistical features of texts also play a role in relatedness estimation. Thus, the ensemble of statistical text features, word embedding models, and models trained on question-answer pairs can achieve good results.

Although the results for Hindi did not match the superior outcomes of other supervised models, such as `mpnet-base` and `LaBSE`, which were trained with semantic labels or similarity scores, the ensemble system's performance still surpasses that of the multilingual BERT embedding model and the `multi-qa` model, both of which were utilized for ensemble learning as base models.

4.1 Biased Performance

We also observe that the unsupervised results for Hindi are not comparable with those from Spanish and English though with the same ensemble approach. This discrepancy stems from the suboptimal performance of the sub-models used in the unsupervised ensemble. For example, the multi-qa-MiniLM model utilized for Hindi only achieves a correlation of 0.466, and the multilingual BERT for Hindi is also less effective compared to the other two languages.

Apart from Hindi, we also applied the same ensemble method to other non-Indo-European languages in the unsupervised track, yet the results scarcely surpassed 0.60 for the validation dataset, so results of other languages were ultimately not submitted.

The results indicate that some multilingual models are biased towards English and Indo-European languages, and perform less effectively for other languages. This bias may be attributed to imbalanced data during the models' pre-training phase.

5 Conclusion

Our system employs an ensemble approach to estimate semantic relatedness, integrating results from multiple systems: textual statistical features, word embedding models, and sentence representation models. Our findings suggest that semantic relatedness can be deduced from a variety of sources. Although some features (e.g., lexical overlap ratio) may not perform as strongly as models specifically designed to obtain sentence representations, the results demonstrate that these features, when used in a combined manner, can outperform many individual systems and collaboratively achieve a better correlation with human judgment on semantic relatedness.

6 Limitation and Future Work

Constrained by the size of the training data and the availability of pre-trained language models, it is regrettable that we did not offer insights into other Asian and African languages. In future research, studies on low-resource languages will be valuable, including tasks such as data collection, annotation, and pre-training models tailored to these languages.

Acknowledgements

We are very grateful for the assistance and discussions provided by Leander Girrbach and Milan Straka.

[5]Table 3 shows `all-mpnet-base-v2` result for English and `paraphrase-multilingual-mpnet-base-v2` model results for Spanish and Hindi, model details: https://www.sbert.net/docs/pretrained_models.html

[6]Feng et al., 2022

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret, editors. 2012. **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, Montréal, Canada.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Michael Günther, Georgios Mastrapas, Bo Wang, Han Xiao, and Jonathan Geuter. 2023. Jina embeddings: A novel set of high-performance sentence embedding models. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 8–18, Singapore. Association for Computational Linguistics.

Xin Huang, Kye Min Tan, Richeng Duan, and Bowei Zou. 2023. Ensemble method via ranking model for conversational modeling with subjective knowledge. In *Proceedings of The Eleventh Dialog System Technology Challenge*, pages 177–184, Prague, Czech Republic. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Suresh Manandhar and Deniz Yuret, editors. 2013. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Association for Computational Linguistics, Atlanta, Georgia, USA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

Anton Osika, Susanna Nilsson, Andrii Sydorchuk, Faruk Sahin, and Anders Huss. 2018. Second language acquisition modeling: An ensemble approach. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 217–222, New Orleans, Louisiana. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, et al. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *arXiv preprint arXiv:2402.08638*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. Semeval-2024 task 1: Semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word

representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Matthew Raffel and Lizhong Chen. 2023. Implicit memory transformer for computationally efficient simultaneous speech translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12900–12907, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

IUCL at SemEval Task 8: A Comparison of Traditional and Neural Models for Detecting Machine Authored Text

Srikar Kashyap Pulipaka, Shrirang Rajendra Mhalgi,
Joseph Edward Larson, Sandra Kübler
Indiana University
{spulipa,srmhalgi,joelarso,skuebler}@iu.edu

Abstract

Since Large Language Models have reached a stage where it is becoming more and more difficult to distinguish between human and machine written text, there is an increasing need for automated systems to distinguish between them. As part of Sem-Eval Task 8, Subtask A: Binary Human-Written vs. Machine-Generated Text Classification, we explore a variety of machine learning classifiers, from traditional statistical methods, such as Naïve Bayes and Decision Trees, to finetuned transformer models, such as RoBERTa and ALBERT. Our findings show that using a finetuned RoBERTa model with optimized hyperparameters yields the best accuracy. However, the improvement does not translate to the test set because of the differences in distribution in the development and test sets.

1 Introduction

Large Language Models (LLMs) are becoming more and more accessible, which has resulted in an increase in machine-generated content across a wide variety of domains, including education, technology, and science. With this increase in machine generated texts from LLMs, and with the increase in the quality of LLM created texts, concerns regarding but not limited to fake product review generation (Adelani et al., 2019) spam/phishing (Weiss, 2019) and fake news generation (Zellers et al., 2019; Brown et al., 2020; Uchendu et al., 2020) have arisen. Weiss (2019) demonstrated that humans can only detect such misuses of LLMs at chance level, which demonstrates the clear need for automated systems to detect machine generated content. In this paper, we describe the IUCL submission to SemEval task 8 (Wang et al., 2024); we focused mostly on comparing traditional and neural models. Our best system ranked 70th out of 137 submissions.

2 Related Work

In terms of impressionistic differences between human generated text and LLM generated text, it has been observed that LLMs tend to be more focused (i.e. less diversion from the subject at hand), more objective, and highly formal. Human texts, on the other hand, are overall more emotional, subjective, and less formal. In terms of linguistic difference, humans use fewer nouns and conjunctions, while employing more punctuation and adverbs. Dependency relations are also shown to be shorter. Lastly, human texts have higher type/token ratios in texts of the same length (Guo et al., 2023) Current LLM models include GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), CTRL (Keskar et al., 2019) and ChatGPT.

We will first begin by discussing statistical approaches to detecting machine-generated content, then using LLM technology itself to do so.

Solaiman et al. (2019) use a bag-of-words approach with TF-IDF feature vectors (both unigrams and bigrams) and a logistic regression model to differentiate between human-written web pages and text generated web pages from GPT2. They examine a different number of parameters of the LLM (117M, 345M, 762M and 1,542M) as well as different sampling methods (k-sampling, p-sampling and pure sampling). This is because an assumption that many researchers take is that language models sample from the head to generate natural looking text e.g. max sampling (Gu et al., 2017) and k-max sampling (Fan et al., 2018). Their findings are that the larger the LLM, the harder to detect how machine-like the generated text is and k samples are easier to detect than pure samples, probably due to the fact that k samples over-produce common words, which is easy to detect using statistical methods.

Gehrmann et al. (2019) use BERT and a group of statistical features: the probability of each word, ab-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1026–1031

solute rank of each word, and entropy of the distribution, and create a tool for users to see specifically what features are more likely to be machine generated over human generated. They clearly show that the model GPT-2 oversamples certain words; it is worth pointing out, however, that as LLMs grow more sophisticated, such methods may not work as well.

Solaiman et al. (2019) use finetuning on RoBERTa and find that it can detect text generated from GPT-2 with an accuracy of 95%. The RoBERTa detector has also been used in detecting fake news articles from several LLMs (Uchendu et al., 2020), Amazon product reviews (Adelani et al., 2019), and biomedical texts (Rodriguez et al., 2022).

3 Data

We used the M4 dataset (Wang et al., 2023) provided by the SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection. We used the English data provided for Subtask A, the Monolingual (English) binary classification task.

The dataset for this subtask consists of 119,757 samples of human-written and machine generated text. There are an additional 5,000 samples as a development set. The test set consists of 34,272 samples.

About 53% of the samples in the training set are machine generated while the rest are human written. The machine generated text was produced by a range of models: ChatGPT and DaVinci by OpenAI, Dolly by Databricks, Cohere. The sources from which the human texts are taken are Reddit, WikiHow, ArXiV, Wikipedia and PeerRead. In contrast, the development set consists of an equal ratio of human and machine generated samples. The machine generated samples are entirely from the Bloomz model. The human sources are also equally distributed between WikiHow, Wikipedia, Reddit, ArXiV and PeerRead. In the test set, 52.5% of the texts are machine generated with GPT4, Cohere, ChatGPT (GPT3.5), Bloomz, Dolly, and DaVinci as sources. Note that this means optimizing a system on development data is difficult since the test data are much closer to the training data than the development data.

Further details about the data and the task are available at the overview of the shared task (Wang et al., 2024).

We present a comparison of a range of classifiers (see below). For those experiments, we use the development set of 5,000 samples for benchmarking and finetuning the model performance.

4 Methods and Features

4.1 Features

Ratio features We started with extraction of features from the dataset that cannot be controlled consciously by authors: stopword ratio and average sentence length. We used the NLTK stopwords[1] (Bird et al., 2009) to calculate the stopword ratio for the dataset. The left graph in Figure 1 shows the distribution for the sentences generated from different sources. The median stopword ratio for humans and different models are around 0.40. It is difficult to distinguish human text from machine text as the distributions of the texts generated by machines are similar to those of the human generated texts. We then computed the average sentence length generated by different sources, see the right graph in Figure 1. The average number of the sentences generated in each of the category is around 21. Again, there is little difference between machine and human generated texts.

Textual features We also used TF-IDF and word unigram features.

4.2 Statistical Learning Methods

We used the ratio features to train Multinomial Naïve Bayes, Random Forest, XGBoost, Logistic Regression and SVC models on the data. For the textual features, we trained SVC, Decision Tree, Logistic Regression and Random Forest classifier models. For all models, we used the scikit-learn implementations (Pedregosa et al., 2011).

We chose the Naïve Bayes classifier because of its simplicity and the ability to handle missing data values. Support Vector Classifier is better at handling high dimensional spaces and is robust to overfitting. Random Forest is an ensemble learning method which is robust to overfitting and provides feature importance ranking, helping to identify the most influential features. Logistic Regression and Multinomial Naïve Bayes classifiers are easy to interpret and are computationally efficient. XGBoost provides a gateway to handle data in a highly efficient and scalable manner. Because of time constraints, we did not perform any hyperparameter

[1] https://gist.github.com/sebleier/554280

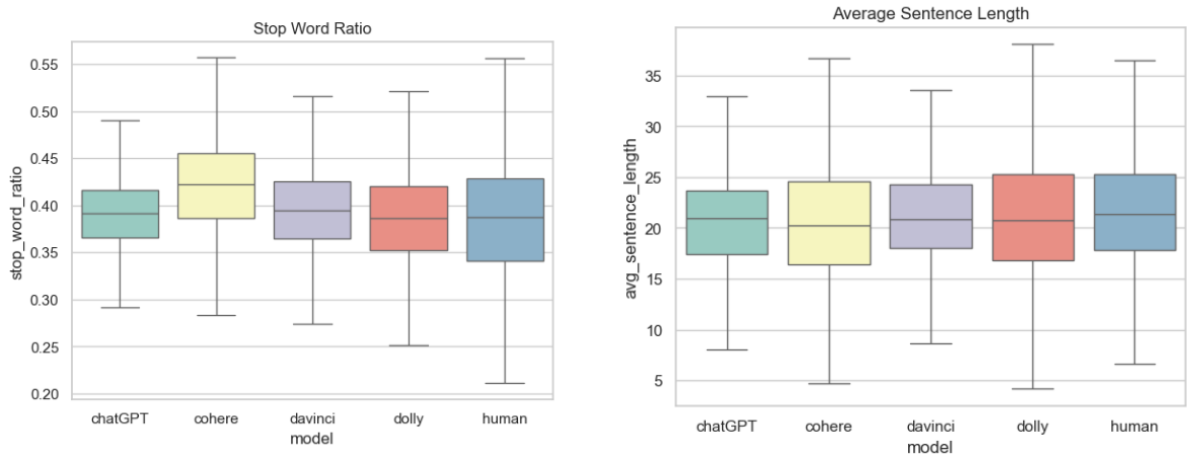


Figure 1: Stop word ratio (left) and average sentence ratio (right) generated by different models.

tuning, and used the default settings to train the models.

4.3 Deep Learning Methods

Fully connected DL model We used the same data preprocessing techniques described in Section 4.1 and trained a fully connected 2 layer neural network having 512 hidden units with ReLU activation. We used the Binary Crossentropy to calculate the loss and Adam optimizer to train our neural network on 100 epochs. We set the batch size to 2048, due to processing limitations and kept a learning rate of 0.001 with an early stopping mechanism in place.

Finetuned Language Models We also finetuned the following language models: BERT and its derivative models ROBERTa and ALBERT. We use the Hugging Face library (transformers) for this task.

BERT (Bidirectional Encoder Representations from Transformers) is a language model developed by Devlin et al. (2019). It is a bidirectional model that uses a transformer architecture. We use the BERT base model for our experiments.

RoBERTa is a variant of BERT developed by Liu et al. (2019). It is pre-trained on a larger corpus of texts. We use the RoBERTa base as well as large models for our experiments. The best performing model of our study is a RoBERTa base model. ALBERT is a smaller version of BERT developed by (Lan et al., 2020). The hyperparameters selected are shown in Table 1.

	RoBERTa	BERT	ALBERT
Learning Rate	5e-5	2e-5	2e-5
Batch Size	8	32	16
Nr. Epochs:	3	4	4
Grad. Acc. St.	4	2	2

Table 1: Hyperparameters for the neural models

5 Results

We will first discuss our results on the development data, then the official results of the shared task.

5.1 Results on the Development Set

The shared task provides a baseline accuracy of 74% using a RoBERTa model. Our aim is to investigate a range of models and features and incrementally improve models, starting out with traditional machine learning models and then moving on to deep learning models.

Table 2 shows the performance of the different combinations of models and features on the development set.

We first look at the statistical methods combined with the standard sparse features, bag of words, and TF-IDF weighted bag of words features. The results in the first block show that the TF-IDF weighted feature results in a lower accuracy than standard frequency counts (56.44% vs. 60.22%) for logistic regression. For this reason, we decided to concentrate on frequency counts. Among the different statistical classifiers, logistic regression reaches the highest results (60.22%), followed by XGBoost with 59.26%.

When we use the ratio features, i.e., stop word

Features	Model	Acc.
TF-IDF	Logistic Regression	56.44
words	Random Forest	58.86
	Naïve Bayes	50.54
	XGBoost	59.26
	Logistic Regression	60.22
Ratio features	Logistic Regression	67.14
BERT	Logistic Regression	63.48
	Fully connected NN	67.19
	Fully connected NN (optimized)	70.11
ALBERT	ALBERT	66.78
RoBERTa	RoBERTa BASELINE	*74.00*
	XLM-RoBERTa Large	77.67
	XLM-RoBERTa (10,000 training samples)	78.24
	XLM-RoBERTa Base Default	79.61
	XLM-RoBERTa Base (optimized)	**79.90**

Table 2: Model comparison with respect to features and accuracy for Dev Set

ratio and average sentence ratio, combined with logistic regression, we reach an accuracy of 67.14%, which is surprising in that this outperforms word features by almost 6% absolute, even though they did not show large differences in Figure 1.

Next, we investigate whether using BERT embeddings instead of sparse features improves results. When we use those features with logistic regression, results increase by 3% absolute to 63.48%, combining them with the fully connected neural network, we reach an accuracy of 70.11%, outperforming the ratio features, but not reaching the baseline provided by the shared task.

We then move on to use BERT and its variants. We start off with ALBERT, a smaller version of BERT. This model gives us an accuracy of 66.78%. This shows that we need a large scale model for good performance. We find that the XLM-RoBERTa model, a multilingual pre-trained model performs better than a RoBERTa model. An XLM-RoBERTa model with full data and default parameters gives us an accuracy of 79.61%. We add gradient accumulation to the finetuning process to speed up training and improve performance. We also reduce the batch size and adjust the learning rate, to get an incremental 0.3% improvement due to the hyperparameters. Optimizing hyperparameters tuning further increases accuracy to 79.90%. This is the best accuracy we have obtained in our experiments. When we compare those results to the XLM-RoBERTa large model with its higher number of parameters, accuracy drops to 77.67%, showing that simply increasing the number of parameters does not guarantee good performance.

A final experiment investigates the importance of the training set size. For this experiment, we reduce the training data to 10,000 samples. This model gives us an accuracy of 78.24%, showing that finetuning XLM-RoBERTa with even a small dataset reaches competitive results. Increasing the training set from 10,000 to about 120,000 results in an increase in accuracy of 1.66% absolute.

System	Score	Rank
Our submission	74.96	70
Baseline	88.46	–
safeai	96.88	1

Table 3: Official Results (accuracy).

5.2 Official Results

We generated our final predictions using the finetuned XLM-RoBERTa system. We show our results in comparison to the best system and the baseline in Table 3. Our submission had an accuracy of 74.96% on the test set and was ranked 70 out of 137 teams. The best ranking team had an accuracy of 96.88%. Note that while our system improved over the baseline for the development data, this is not the case for the test data. This is most likely a consequence of the different distributions between the development and test data.

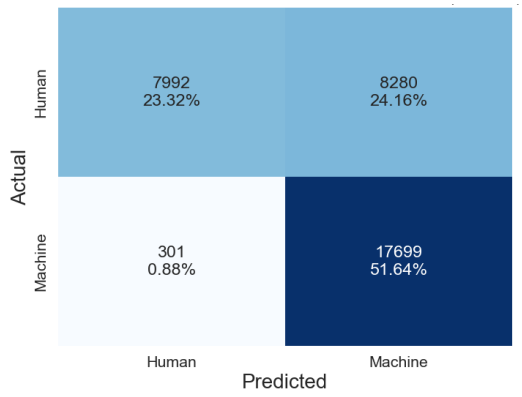


Figure 2: Confusion Matrix of the best model (Test Set)

5.3 Discussion

We had a closer look at the confusion matrix for the best performing model, the optimized XLM-RoBERTa model, on the test data, shown in Figure 2. We notice that the model has a tendency to incorrectly identify human samples as machine generated (false positives) in 8,280 cases, as opposed to just 301 cases of false negatives.

One of the limitations of our work is that we have not explored data processing and augmentation techniques that can help us improve the performance of the model.

6 Conclusion and Future Work

In this project, we have investigated the performance of various machine learning models. We found that our best performing model is a base XLM-RoBERTa model that is fine-tuned on the dataset. Using the smaller ALBERT or the large XLM-RoBERTa models resulted in decreases in accuracy. However, we also see that finetuning is very sensitive to underlying data characteristics, since the gains we saw on the development set did not translate to equivalent gains on the test set.

There is a significant scope for improvement in the performance of the models by working on further text preprocessing and feature engineering. Future work includes using ensemble methods that combines the finetuned models along with a model using ratio features.

Acknowledgements

This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2019. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. arXiv.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020*, volume 33, pages 1877–1901, Vancouver, Canada.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, MN.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 889–898, Melbourne, Australia.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy.

Jiatao Gu, Kyunghyun Cho, and Victor O.K. Li. 2017. Trainable greedy decoding for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1978, Copenhagen, Denmark.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. In *arXiv*, 2301.07597.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. In *arXiv*, 1909.05858.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the Eighth International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. In *arXiv*, 1907.11692.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI blog.

Juan Diego Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233, Seattle, United States.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. In *arXiv*, 1908.09203.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Max Weiss. 2019. Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions. *Technology Science*, 2019121801.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA.

RACAI at SemEval-2024 Task 10: Combining algorithms for code-mixed Emotion Recognition in Conversation

Sara Niță and **Vasile Păiș**

Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy

Bucharest, Romania

saramaria.nita9@gmail.com, vasile@racai.ro

Abstract

Code-mixed emotion recognition constitutes a challenge for NLP research due to the text's deviation from the traditional grammatical structure of the original languages. This paper describes the system submitted by the RACAI Team for the SemEval 2024 Task 10 - EDiReF subtasks 1: Emotion Recognition in Conversation (ERC) in Hindi-English code-mixed conversations. We propose a system that combines a transformer-based model with two simple neural networks.

1 Introduction

Emotion recognition in conversation (ERC) (Kumar et al., 2023) is a crucial task in conversational artificial intelligence research that aims to identify the emotion of each utterance in a conversation. ERC proves useful in applications such as opinion mining and empathetic dialog systems. However, many of the existing models and datasets for emotion recognition are single-language. But, proliferating mixed language interactions have boosted interest in code-mixed natural language processing (NLP) tasks.

The present work describes the system that participated in the shared task "Emotion Discovery and Reasoning its Flip in Conversation (EDiReF)", task 10, organized at SemEval 2024 (Kumar et al., 2024). The EDiReF shared task is made up of three subtasks: (i) Emotion Recognition in Conversation (ERC) in Hindi-English code-mixed conversations, (ii) Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations, and (iii) EFR in English conversations. Out of these subtasks, our team participated only in sub-task (i).

Many current approaches for diverse NLP tasks, including ERC, relies on the application of large language models (LLMs) and fine-tuning them on a specific dataset. For this work we were interested in determining how existing language resources, such as emotion lexicons, could be used to complement and improve the predictions of LLM-based approaches. For this reason, our final system, as detailed in Section 4.2, is an ensemble of a BERT-based implementation and traditional feature-based approaches, employing an emotion lexicon. Apart from the emotion lexicon, we did not use any external datasets. Only the dataset provided by the task organisers was used as an emotion annotated dataset. We also took into account the requirement expressed by the task organizers, that no data from task 2 or task 3 can be used to train/evaluate task 1.

This paper is organized as follows: Section 2 provides related work, Section 3 briefly presents the task and describes the dataset, Section 4 gives an overview of the participating system, including pre-processing and architecture, Section 5 presents the results, and Section 6 gives conclusions.

2 Related work

Wang et al. (2020) recognizes the importance of ERC for developing empathetic machines in a variety of areas. The authors model the ERC task as sequence tagging where a Conditional Random Field (CRF) layer is leveraged to learn the emotional consistency in the conversation. Experiments are performed on three datasets: IEMOCAP (Busso et al., 2008), DailyDialogue (Li et al., 2017), and MELD (Poria et al., 2019). The authors ackowledge an imbalanced data distribution in some of the ERC datasets, similar to the distribution provided for the current task (as described in Section 3).

Ghosal et al. (2019) propose Dialogue Graph Convolutional Network (DialogueGCN), a graph neural network based approach to ERC. The authors test the approach on a number of datasets, including IEMOCAP and MELD, showing good results.

Song et al. (2022) employ a Supervised Prototypical Contrastive Learning (SPCL) loss for the ERC task. In this case, the SPCL aims to solve the imbal-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1032–1037

anced classification problem through contrastive learning. Their approach further improve results on the IEMOCAP and MELD datasets, achieving F1 scores of 69.74% and 67.25%, respectively.

De Bruyne et al. (2022) evaluate the language-dependence of an mBERT-based emotion detection model. Experiments included the Hindi and English languages. Their findings suggest that there could be evidence for the language-dependence of emotion detection performance.

Datasets and systems for emotion recognition have been proposed for other languages as well. For example, for the Romanian language, Ciobotaru and Dinu (2021) introduced the RED dataset for emotion detection in Romanian tweets. Colhon et al. (2016) showed that particular Romanian language words, such as negations, intensifiers and diminishers, affect the detected polarity of the sentiments described in natural language texts. Furthermore, Tăiatu et al. (2023) introduced RoBERTweet, a BERT-like LLM for Romanian language. The authors also describe a system using the RoBERTweet model for emotion detection outperforming previous general-domain Romanian and multilingual language models.

Laki and Yang (2023) explore sentiment analysis with neural models for the Hungarian language. The authors try to solve the class imbalance problem either by removing examples from the highly represented class (while keeping the same number of examples as the least represented class) or by duplicating examples from the least represented class. In addition they explore data augmentation by means of machine translation and cross-lingual transfer. Different Hungarian language LLMs, especially BERT-like LLMs, are considered for the experiments. Üveges and Ring (2023) introduce HunEmBERT, a fine-tuned BERT-like model for classifying sentiment and emotion in political communication in the Hungarian language.

Apart from neural network models and datasets, lexicons constitute another type of useful resources for sentiment analysis. This type of resources have been created for different languages. Lupea and Briciu (2019) introduced the Romanian Emotion Lexicon (RoEmoLex v.3). It contains associations between a series of words and eight basic emotions (Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise, Trust) and two sentiment orientations (Positivity and Negativity). Initially translated from an English version, it now contains additional tags, including derived emotions, part-of-speech, additional polarity scores and conceptual category information. It was also expanded with synonyms of the original terms and new words and phrases.

Mohammad and Turney (2010, 2013) propose the NRC Word-Emotion Association Lexicon (EmoLex). It contains English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done by crowdsourcing. The authors assess that despite some cultural differences, the majority of the affective norms are stable across languages. Thus, the lexicon is also provided in over 100 languages by automatic translating the English terms using Google Translate.

Various datasets for sentiment classification, including those mentioned in this section, suffer from a class imbalance problem. Frameworks for data augmentation, such as NL-Augmenter (Dhole et al., 2023), have been proposed, allowing automatic enrichment of less represented classes. Chawla et al. (2002) proposed SMOTE, a synthetic minority over-sampling technique. Their approach combines under-sampling of the majority class with a special form of over-sampling the minority class. The minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with replacement, thus reducing the potential overfitting.

3 Dataset and task

The goal of the emotion recognition task is to classify a given sentence from a dialogue into one of eight emotion states: the seven universal human emotions as described by Dr. Paul Ekman (Ekman, 1992) ("anger", "surprise", "contempt", "disgust", "fear", "joy", "sadness") and "neutral". The dataset files, with splits for training, validation, and testing, were provided in JSON format. The records contain fields for the name of the episode the lines were taken from, a list of speakers, the actual dialogue (list of sentences called "utterances"), and a list with the emotions attributed to each line ("emotions" or "labels"). The utterances included some unrecognized characters that needed to be removed. The training dataset contains 343 entries (8,506 utterances), the validation dataset contains 46 entries (1,354 utterances), while the test dataset contains 57 entries (1,580 utterances).

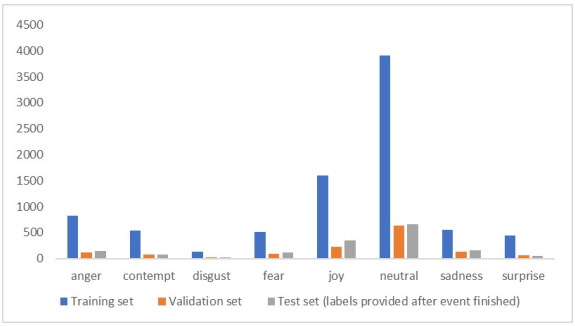


Figure 1: Emotion distribution for train, validation and test sets

The labels distribution for the train, validation and test splits is given in Figure 1. Similar to other emotion recognition datasets, such as IEMOCAP or MELD, as reported in Section 2, there is a class imbalance present in the task dataset as well. Many sentences are marked as being "neutral", while the next class, considering the number of samples, is "joy". The least represented class is "disgust".

4 Methodology

4.1 Pre-Processing

In the pre-processing stage, all blank characters, including new lines, tabs, and other unrecognized UTF-8 characters, were transformed into regular spaces. Dialogues were split into individual sentences and duplicates removed from the training set.

Given the observation of De Bruyne et al. (2022) regarding the possible language-dependence of emotion detection performance, combined with the existence of a large number of emotion lexicons in the English language, individual sentences were completely translated into English, removing any Hindi text (including roman script). For this purpose, we employed the GoogleTranslator from the deep_translator library.

4.2 Overall system architecture

The system is comprised of two parts: one being a multilingual BERT LLM and the other consisting of a Decision Tree and a Random Forest classification algorithms, employing additional features.The final result was obtained by running the three sets of predictions from the models through a voting system. If two or all three models predict the same emotion, then this becomes the final prediction, but if they each give different results, then the BERT prediction is chosen as the final prediction, because when taken separately, BERT has better results then either decision trees or random forest, as shown in in Table 1. A diagram of the entire system is given in Figure 2.

4.3 Decision Tree and Random Forest

To aid in feature construction, the text was lemmatized by employing the WordNet (Fellbaum, 1998) lemmatizer available in the NLTK library[1].

From the translated sentences a set of handcrafted features were produced, some of which were binary features associated with each one of the seven emotions. Through the use of an English lexicon, the NRC-Emotion-Lexicon-Wordlevel[2] (Mohammad and Turney, 2013), the feature was either marked as "1", if the emotion was the most commonly found one among the meanings of the words in a given sentence, or as "0". The lexicon unfortunately did not contain any data about the “contempt” value of words. As a unified resource for both Hindi and English was not successfully found, the translation previously performed was necessary. The other features were: length, the number of sentences in an utterance, punctuation (for full stop, question mark, exclamation mark and ellipsis), ratio of words from the lexicon that were predominantly positive or negative and the confidence of the lexicon. The confidence was computed based on the number of words belonging to different classes which were found in the lexicon for a given sample.

For the decision tree predictions only, new examples were synthesized using a SMOTE pipeline (Chawla et al., 2002) due to the imbalanced nature of the dataset.

4.4 BERT

The LLM used for training the system was bert-base-uncased. This was chosen due to our assumption that a smaller model may benefit more from additional resources, such as an emotion lexicon. The LLM classifier has two additional linear layers, with 2,048 and 1,024 cells respectively, employing ReLU and tanh activation functions respectively. These are followed by a final class prediction head. The model was trained for at least 5 epochs and a maximum of 20 epochs, with early stopping, when there was no improvement for 3 epochs. During the first 3 epochs, the LLM was frozen and only

[1] https://www.nltk.org/

[2] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

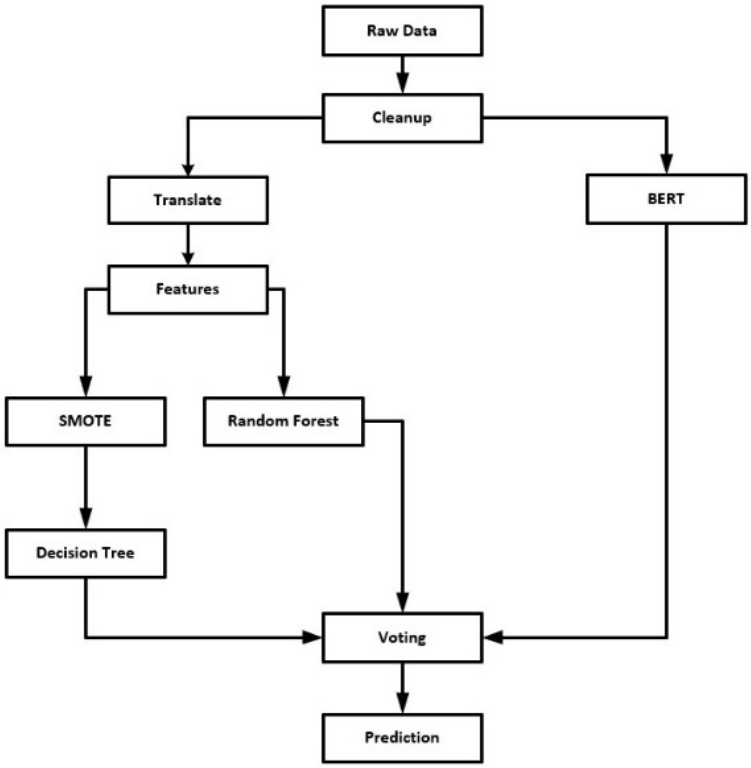


Figure 2: System architecture.

the last linear layers were actually trained. A batch size of 6 was used. The learning rates for the LLM and the other layers were kept separated. The best hyper-parameters were found to be encoder learning rate 1.0e-05, and linear layers learning rate 3.0e-05. The final model training lasted 18 epochs.

5 Results and discussion

Results are given in Table 1 for the test dataset, in terms of weighted precision, recall, and F1 scores. In this case, the weighted recall is equal to the accuracy measure. The baseline is computed on the assumption that all results are neutral. As expected, due to the class imbalance, this provides the best accuracy. Decision tree and random forest classifiers provide results worse than BERT alone and even worse than the baseline approach. This translates into words not being found in the lexicon or words that may mean different things in context, while the lexicon does not take into account the context. Even though the voting mechanism favors the BERT prediction, it seems it actually decreases all the metrics. It is however worth observing the precision score associated with the random forest classifier employing features generated based on the lexicon which is quite high (only 4% under the precision offered by the LLM predictor).

System	P	R	F1
BERT	**36.2**	37.6	**35.0**
DT	7.8	16.7	10.5
RF	0.326	16.7	18.2
Voting	35.2	33.9	30.9
Baseline	17.1	**0.42**	0.24

Table 1: Results on the test dataset.

6 Conclusion and future work

The proposed system tried to combine a lexicon approach with a LLM prediction, considering that a manually created emotion lexicon could complement the LLM predictions. Nevertheless, even though the precision given by the random forest classifier based on features derived using the lexicon is surprisingly good, the recall is significantly lower, thus resulting in an overall lower F1 score, even in the face of a LLM with a reduced number of parameters.

In accordance with open science principles, the code for the described system is made available

open source in its own GitHub repository[3].

The class imbalance problem was tackled only with the SMOTE technique. However, as mentioned in Section 2, different frameworks for data augmentation are available. Future work may include experiments with other data augmentation techniques for the minority classes.

As documented in Section 2, different authors have shown improvements using language-specific and domain-specific LLMs. For this work, we focused on a single BERT LLM. Other LLMs, with more specificity or a larger number of parameters, may provide better results. However, the question regarding the possible enhancement of predictions using additional resources, such as emotion lexicons, still remains valid.

Limitations

The current system implementation makes use of English-only emotion lexicons. The system architecture does not take into account long messages that surpass the direct capability of the LLMs used.

Ethics Statement

We do not foresee ethical concerns with the research presented in this paper. However, it is important to acknowledge that unintended bias might be present in the dataset and this could be reflected in the resulting models. Furthermore, since the emotion lexicons have been created by people they capture various human biases which may be reflected in the final system.

References

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. 2008. Iemocap: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

Alexandra Ciobotaru and Liviu P. Dinu. 2021. RED: A novel dataset for Romanian emotion detection from tweets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 291–300, Held Online. INCOMA Ltd.

[3] https://github.com/SaNita9/ediref2024-subtask-1

Mihaela Colhon, Mădălina Cerban, Alex Becheru, and Mirela Teodorescu. 2016. Polarity shifting for romanian sentiment classification. In *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–6.

Luna De Bruyne, Pranaydeep Singh, Orphee De Clercq, Els Lefever, and Veronique Hoste. 2022. How language-dependent is emotion detection? evidence from multilingual BERT. In *Proceedings of the The 2nd Workshop on Multi-lingual Representation Learning (MRL)*, pages 76–85, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Kaustubh Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahadiran, Simon Mille, Ashish Shrivastava, Samson Tan, Tongshang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondřej Dušek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Tanya Goyal, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honoré, Ishan Jindal, Przemysław K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxine Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Meunnighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl, Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Păiș, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicholas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Yiwen Shi, Haoyue Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn, Athena Wang, Zijian Wang, Zijie J. Wang, Gloria Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyu Wu, Witold Wydmanski, Tianbao Xie, Usama Yaseen, Michael A. Yee, Jing Zhang, and Yue Zhang. 2023. NL-Augmenter: A framework for task-sensitive natural language augmentation. *Northern European Journal of Language Technology*, 9(1):1–41.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 154–164, Hong Kong, China. Association for Computational Linguistics.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

László Laki and Zijian Yang. 2023. Sentiment analysis with neural models for hungarian. *Acta Polytechnica Hungarica*, 20:109–128.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Mihaiela Lupea and Anamaria Briciu. 2019. Studying emotions in romanian words using formal concept analysis. *Computer Speech & Language*, 57:128–145.

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA. Association for Computational Linguistics.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Xiaohui Song, Longtao Huang, Hui Xue, and Songlin Hu. 2022. Supervised prototypical contrastive learning for emotion recognition in conversation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5197–5206, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Iulian-Marius Tăiatu, Andrei-Marius Avram, Dumitru-Clementin Cercel, and Florin Pop. 2023. Robertweet: A bert language model for romanian tweets. In *Natural Language Processing and Information Systems*, pages 577–587, Cham. Springer Nature Switzerland.

Yan Wang, Jiayu Zhang, Jun Ma, Shaojun Wang, and Jing Xiao. 2020. Contextualized emotion recognition in conversation as sequence tagging. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 186–195, 1st virtual meeting. Association for Computational Linguistics.

István Üveges and Orsolya Ring. 2023. Hunembert: A fine-tuned bert-model for classifying sentiment and emotion in political communication. *IEEE Access*, 11:60267–60278.

ROSHA at SemEval-2024 Task 9: BRAINTEASER A Novel Task Defying Common Sense

Mohammadmostafa Rostamkhani, Shayan Mousavinia, Sauleh Eetemadi
Iran University of Science and Technology
{mo_rostamkhani97, sh_mousavinia}@comp.iust.ac.ir, sauleh@iust.ac.ir

Abstract

In our exploration of SemEval 2024 Task 9, specifically the challenging BRAINTEASER: A Novel Task Defying Common Sense, we employed various strategies for the BRAINTEASER QA task, which encompasses both sentence and word puzzles. In the initial approach, we applied the XLM-RoBERTa model both to the original training dataset and concurrently to the original dataset alongside the BiRdQA dataset and the original dataset alongside RiddleSense for comprehensive model training. Another strategy involved expanding each word within our BiRdQA dataset into a full sentence. This unique perspective aimed to enhance the semantic impact of individual words in our training regimen for word puzzle (WP) riddles. Utilizing ChatGPT-3.5, we extended each word into an extensive sentence, applying this process to all options within each riddle. Furthermore, we explored the implementation of RECONCILE (Round-table conference) using three prominent large language models—ChatGPT, Gemini, and the Mixtral-8x7B Large Language Model (LLM). As a final approach, we leveraged GPT-4 results. Remarkably, our most successful experiment yielded noteworthy results, achieving a score of 0.900 for sentence puzzles (S_ori) and 0.906 for word puzzles (W_ori).

1 Introduction

Human reasoning involves two primary types of thinking: vertical and lateral. Vertical thinking, synonymous with linear, convergent, or logical thinking, follows a sequential analytical process based on rationality and rules. Conversely, lateral thinking, often referred to as "thinking outside the box," is a divergent and creative process that challenges preconceptions by approaching problems from new perspectives. Despite the success of language models in tasks requiring implicit and complex reasoning, there is a notable lack of attention to lateral thinking puzzles within the NLP community. To address this gap, the BRAINTEASER Question Answering task (Jiang et al., 2023) has been introduced, designed to evaluate a model's ability to exhibit lateral thinking and challenge default commonsense associations. SemEval 2024 Task 9, BRAINTEASER (Jiang et al., 2024b) comprises two subtasks, Sentence Puzzle and Word Puzzle, which require unconventional thinking to overcome commonsense "defaults" without violating hard constraints. An adversarial subset is included in both tasks, created by manually modifying original brain teasers without altering their underlying reasoning paths. In our initial series of experiments, our focus is on fine-tuning XLM-RoBERTa (Conneau et al., 2020) in three variations: once on the original training data, once alongside the BiRdQA dataset (Zhang and Wan, 2022), and once alongside the RiddleSense dataset (Lin et al., 2021). Additionally, we introduced an innovative approach involving the extension of each word in the BiRdQA dataset into a complete sentence. This method aims to enhance the contextual meaning of individual words during the training process for word puzzle (WP) riddles. To achieve this, we utilized ChatGPT-3.5 to expand each word into a comprehensive sentence, applying this transformation to all options within each riddle. Subsequently, our exploration extends to the application of RECONCILE (Round-table conference) (Chen et al., 2023), incorporating three substantial language models: GPT 3.5, Gemini, and the Mixtral-8x7B (Jiang et al., 2024a) Large Language Model (LLM), a pre-trained generative Sparse Mixture of Experts. Noteworthy is the superior performance of the Mixtral-8x7B model compared to Llama 2 70B across various benchmarks. In the third set of experiments, we assess the zero-shot performance of GPT-4 using the Copilot GUI. Our observations highlight a significant superiority of GPT-4 over alternative models and methods. Furthermore, our

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1038–1042

findings underscore the collaborative utilization of Large Language Models (LLMs) in a round-table format, showcasing substantial enhancements in overall performance. Evaluation metrics are based on two accuracy measures: Instance-based accuracy, treating each question (original/adversarial) as a distinct instance, and group-based accuracy, where each question and its associated adversarial instances form a group, and a system is awarded a score of 1 only if it correctly solves all questions within the group. Our submission to the evaluation phase comprised XLM-RoBERTa fine-tuned on the original training dataset and BiRdQA dataset. The resulting method ranked 25 out of 31 in sentence puzzles and 20 out of 23 in word puzzles. For a detailed implementation of our method, refer to our GitHub repository.

2 Background

The model's inputs consisted of the puzzle and its corresponding choices, provided as input to XLM-RoBERTa. For alternative methods, we employed a prompt, feeding both the puzzle and choices to the model. All puzzles were written in English. To enhance the training of XLM-RoBERTa, we augmented the primary training dataset with additional datasets, namely BiRdQA and RiddleSense. In the context of word puzzles, we further enriched each choice by transforming it into a complete sentence using ChatGPT. The output from all models and methods was expressed as a numerical representation, denoting the correct choice in a zero-based format.

3 System overview

3.1 Preprocessing

In the preprocessing stage, we employ the following steps for the XLM-RoBERTa model: Each choice is concatenated with the corresponding question and subsequently tokenized. In the case of the BiRdQA and RiddleSense datasets, each riddle initially contains 5 options. However, the standard format, based on data validation, necessitates 4 options. To handle this, we transform each riddle into two separate riddles. The approach involves first removing the correct answer from the set of 5 options, resulting in 4 shuffled options. We then create two new riddles from this set by selecting 3 options for each. Finally, we add the correct answer back to the list of labels for each of the new riddles. In an alternative approach, we endeavored to transform each word into a sentence for every option within the BiRdQA dataset. This strategy aimed to enhance the robustness of our model, facilitating a more comprehensive understanding of each option. The rationale behind this was rooted in the notion that comprehending a sentence is generally more straightforward than understanding an isolated word. To execute this transformation, we presented each option to ChatGPT-3.5 with the prompt: "What is the definition of "text"? Write in a sentence." This process generated an extensive file resembling a dictionary. Throughout our training procedure, instead of utilizing individual words, we incorporated the respective definitions created by ChatGPT into our model. For methods utilizing Large Language Models (LLMs), no specific preprocessing is applied. Instead, we use the data in the format of our prompt without any additional preprocessing steps.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	1×10^{-5}
Epochs	10
Batch size	4
Scheduler	Cosine Annealing
Loss Function	Categorical Cross Entropy

Table 1: Values of hyperparameters

3.2 Dataset

To construct the dataset for the XLM-RoBERTa model, we store tokenized sentences for each choice, concatenated with the corresponding question, and include the corresponding label indicating the correct answer to the riddle. Additionally, we incorporate the BiRdQA dataset, designed for bilingual question answering on challenging riddles, and the RiddleSense dataset, alongside the original training dataset. The creation of new datasets from these sources is detailed in the preprocessing section. The original train and test datasets for sentence puzzles comprise 507 and 120 instances, respectively. For word puzzles, the train and test datasets consist of 396 and 96 instances, respectively. The original RiddleSense and BiRdQA datasets initially contain 3510 and 4093 instances, and after applying the transformations outlined in the preprocessing section, they expand to 7020 and 8186 instances, respectively.

DataSet	W_ori	W_sem	W_con	W_ori_sem	W_ori_sem_con	W_overall
Original Dataset	0.438	0.469	0.438	0.344	0.188	0.448
Original + BiRdQA	**0.625**	0.469	**0.469**	0.468	0.281	**0.521**
Original + RiddleSense	0.531	**0.562**	0.438	**0.5**	**0.375**	0.51
Original + BiRdQA (Word Extender)	0.468	0.468	0.25	0.406	0.125	0.375

Table 2: Results of fine-tuned models

Round	Model	S_ori	S_sem	S_con	S_ori_sem	S_ori_sem_con	S_overall
Round 1	**ChatGPT**	0.575	0.700	0.475	0.525	0.300	0.583
	Gemini	0.750	0.750	0.675	0.675	0.575	0.725
	Mixtral-8x7B	0.725	0.625	0.600	0.600	0.450	0.650
Round 2	**ChatGPT**	0.625	0.725	0.700	0.525	0.450	0.683
	Gemini	0.750	0.775	**0.725**	**0.700**	**0.600**	0.750
	Mixtral-8x7B	0.700	0.725	0.600	0.625	0.450	0.675
Round 3	**ChatGPT**	0.700	0.725	0.650	0.625	0.550	0.692
	Gemini	**0.775**	**0.800**	0.700	**0.700**	0.550	**0.758**
	Mixtral-8x7B	0.725	0.650	0.525	0.625	0.375	0.633
Round 4	**ChatGPT**	0.650	0.750	0.675	0.600	0.525	0.692
	Gemini	0.725	**0.800**	0.650	0.650	0.525	0.725
	Mixtral-8x7B	0.675	0.725	0.575	0.625	0.450	0.658

Table 3: Results of Round-Table on sentence puzzle

3.3 Model

We opted for XLM-RoBERTa as our model for this problem due to its pre-training on 100 different languages, indicating a robust understanding of language. Our fine-tuning process involved updating all the model weights using gradient descent on datasets we created. The architecture includes a multiple-choice head with 4 choices over the XLM-RoBERTa model, and we apply Categorical Cross-Entropy loss. For implementing the RECONCILE method, we leverage GPT-3.5, Gemini, and the Mixtral-8x7B Large Language Model (LLM), with certain adaptations to the original method designed for binary classification. We modified it to suit multiple-choice questions and incorporated 4 rounds for our specific application. In each round, the model is prompted to think step by step (Zhou et al., 2023), generating the correct answer and providing a confidence level (0 to 100) along with a reasoning for the selected choice. The original authors suggested that 4 rounds are sufficient for convergence. The output of all models from the previous round serves as input for the next round, where the model evaluates its logical consistency. No fine-tuning is applied to this method. When utilizing GPT-4 with the Copilot interface, we prompt the model to think step by step and generate the correct option. The model provides the corresponding confidence level and a rationale for choosing that particular choice.

4 Experimental setup

We allocated 20% of the original dataset for our validation set, resulting in 80 samples for validation in the word puzzle (WP) domain and 102 samples for validation in the sentence puzzle (SP) domain. Notably, when incorporating additional datasets into our training data, we maintained consistency by retaining the original validation dataset throughout the training process. This decision was driven by the recognition that the supplementary data introduced distinct variations compared to the original training and testing data. Preserving the originality of the validation data aimed to uphold the quality and uniqueness of the final model.
For fine-tuning using XLM-RoBERTa, we utilized the Hugging Face platform and implemented a cosine annealing scheduler.

5 Results

Leveraging the BiRdQA and RiddleSense datasets led to enhancements across all the metrics utilized for evaluating our model, surpassing the performance observed with the original dataset.

The findings presented in table 3 and table

Round	Model	W_ori	W_sem	W_con	W_ori_sem	W_ori_sem_con	W_overall
Round 1	**ChatGPT**	0.375	0.313	0.438	0.219	0.125	0.375
	Gemini	**0.719**	0.594	**0.813**	0.500	**0.438**	**0.708**
	Mixtral-8x7B	0.688	**0.625**	0.469	0.500	0.281	0.594
Round 2	**ChatGPT**	0.5	0.469	0.469	0.406	0.219	0.479
	Gemini	0.656	0.594	0.594	**0.531**	0.375	0.615
	Mixtral-8x7B	0.594	0.563	0.469	0.406	0.188	0.542
Round 3	**ChatGPT**	0.500	0.344	0.469	0.313	0.156	0.438
	Gemini	0.625	0.563	0.625	0.438	0.313	0.604
	Mixtral-8x7B	0.594	0.500	0.531	0.406	0.219	0.542
Round 4	**ChatGPT**	0.500	0.406	0.438	0.375	0.156	0.448
	Gemini	0.594	0.531	0.594	0.438	0.281	0.573
	Mixtral-8x7B	0.500	0.406	0.469	0.344	0.156	0.458

Table 4: Results of Round-Table on word puzzle

Model	S_ori	S_sem	S_con	S_ori_sem	S_ori_sem_con	S_overall	W_ori	W_sem	W_con	W_ori_sem	W_ori_sem_con	W_overall
XLM-RoBERTa (fine-tuned on original dataset)	0.525	0.550	0.625	0.500	0.400	0.567	0.438	0.469	0.438	0.344	0.188	0.448
GPT-4 (Copilot)	**0.900**	**0.875**	**0.825**	**0.875**	**0.775**	**0.867**	**0.906**	**0.875**	**0.875**	**0.844**	**0.719**	**0.885**

Table 5: Comparison between copilot and XLM-RoBERTa results

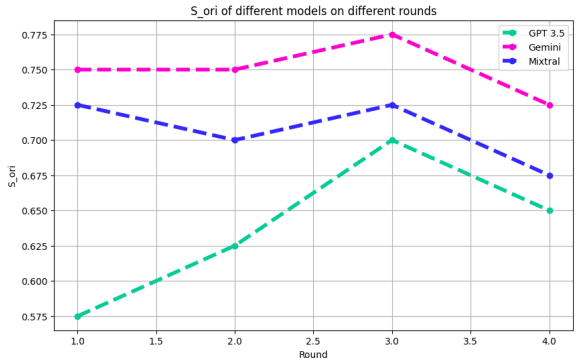


Figure 1: Visualization of Round-Table results for sentence puzzle

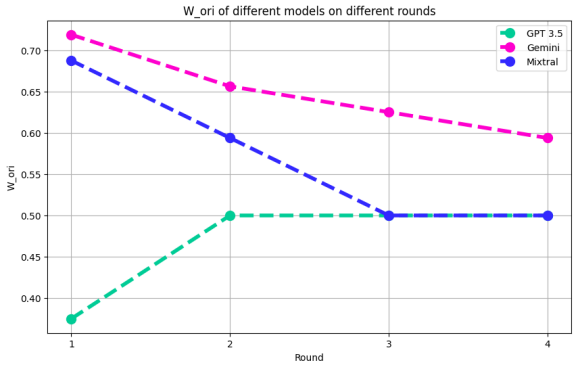


Figure 2: Visualization of Round-Table results for word puzzle

4 indicate that the incorporation of round-table discussions can enhance model performance in sentence puzzles, but conversely, it leads to a decrease in performance for word puzzles. This discrepancy may stem from the fact that, in solving sentence puzzles, some models can provide correct reasoning and influence others positively, whereas the complexity of reasoning in word puzzles may result in incorrect reasoning leading other models astray. Optimal results suggest that employing 3 rounds is most effective for sentence puzzles, while 1 round is preferable for word puzzles. Notably, Gemini consistently outperforms all other models across all rounds. Furthermore, this approach demonstrates its efficacy in boosting the performance of GPT 3.5 in both sentence and word puzzles.

GPT-4 consistently outperformed other models by a significant margin, demonstrating superior results across all metrics.

6 Conclusion

This study explores various methodologies for tackling SemEval 2024 Task 9: "BRAINTEASER: A Novel Task Defying Common Sense." To enhance our model's performance in word puzzles, we incorporate additional datasets for fine-tuning. Additionally, we introduce a modified round-table approach implemented over four rounds. We also evaluate the zero-shot performance of GPT-4 on this task,

Question	Options	BiRdQA	Orginal	RiddleSense	BiRdQA Word Extender	Correct
What kind of stock doesn't have shares?	Small-cap stock, Livestock, Growth stock, None of above	0	0	**1**	2	**1**
What kind of birds always make noise?	Humming bird, Hawk, Owl, None of above	**0**	2	2	1	**0**
What type of chase never involves running?	Escape chase, Paperchase, Risky chase, None of above	0	2	**1**	0	**1**
What kind of tree can you hold in your hands?	Oak, Pine, Palm, None of above	0	0	1	**2**	**2**
What species of geese engages in snake-fighting?	Canada goose, Snow goose, Mongoose, None of above	1	1	1	0	**2**

Table 6: Examples of predictions from different models

which demonstrates superior results across all metrics.

References

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023. Reconcile: Round-table conference improves reasoning via consensus among diverse llms.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024b. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge.

Yunxiang Zhang and Xiaojun Wan. 2022. Birdqa: A bilingual dataset for question answering on tricky riddles.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers.

Sharif-STR at SemEval-2024 Task 1: Transformer as a Regression Model for Fine-Grained Scoring of Textual Semantic Relations

Seyedeh Fatemeh Ebrahimi [♡], **Karim Akhavan Azari**[♡], **Amirmasoud Iravani**[♯]
Hadi Alizadeh[♢], **Zeinab Sadat Taghavi**[♡], **Hossein Sameti**[♡]
Ferdowsi University of Mashhad, Mashhad, Iran[♯]
Sharif University of Technology, Tehran, Iran[♡]
Iran Broadcasting University, Tehran, Iran[♢]

{sfati.ebrahimi, karim.akhavan, zeinabtaghavi, sameti}@sharif.edu
a.iravani@mail.um.ac.ir
alizadeh.hadi08@gmail.com

Abstract

Semantic Textual Relatedness holds significant relevance in Natural Language Processing, finding applications across various domains. Traditionally, approaches to STR have relied on knowledge-based and statistical methods. However, with the emergence of Large Language Models, there has been a paradigm shift, ushering in new methodologies. In this paper, we delve into the investigation of sentence-level STR within Track A (Supervised) by leveraging fine-tuning techniques on the RoBERTa transformer. Our study focuses on assessing the efficacy of this approach across different languages. Notably, our findings indicate promising advancements in STR performance, particularly in Latin languages. Specifically, our results demonstrate notable improvements in English, achieving a correlation of 0.82 and securing a commendable 19th rank. Similarly, in Spanish, we achieved a correlation of 0.67, securing the 15th position. However, our approach encounters challenges in languages like Arabic, where we observed a correlation of only 0.38, resulting in a 20th rank.

1 Introduction

STR delineates the meaningful association between linguistic units, showcasing conceptual proximity within a shared semantic frame (Taieb et al., 2019; Abdalla et al., 2021). For instance, "cup" and "coffee" are related in meaning, yet they are not synonymous (Jurafsky and Martin, 2009). Despite its crucial role in various NLP applications such as Spelling Correction, Word Sense Disambiguation, Plagiarism Detection, Opinion Mining, and Information Retrieval (Franco-Salvador et al., 2016; Chen et al., 2017; Taieb et al., 2019), STR has garnered less attention compared to Semantic Textual Similarity (STS) due to a scarcity of available datasets. Addressing this gap, Abdalla et al. (2021), and Ousidhoum et al. (2024a) contributed to the field by constructing the first sentence-level STR datasets. In this paper, we endeavor to tackle the STR problem within shared Task 1(Ousidhoum et al., 2024b), Track A, leveraging supervised data in English, Spanish, and Arabic languages provided by Ousidhoum et al. (2024a). Additionally, we briefly explore Track C and provide supplementary details in Appendix B as a secondary objective.

Building upon the findings of Abdalla et al. (2021), which underscore the superior performance of fine-tuning Transformer models in supervised tasks, our proposed system captures the relationship among sentences by fine-tuning the RoBERTa Transformer (Liu et al., 2019). At the core of our system, we employ a pre-trained RoBERTa model as a regression model and fine-tune it to generate a floating-point value for the input text. During the pre-training process of RoBERTa, the emphasis is placed on tasks related to NLU. This involves exposing the model to a diverse range of linguistic contexts and training it to comprehend the nuances of language. Furthermore, the integration of a Classifier Head enables sentence classification, a pivotal aspect of our system architecture elaborated upon in section 3.

Our experimental results showcase promising performance on English and Spanish datasets, achieving respective correlation rates of 0.82 and 0.67 on test data, surpassing the baseline correlation set by SemEval-2024 at Subtask A (Ousidhoum et al., 2024b). However, the model's performance on Arabic data falls short, yielding only a 38% correlation on development data. We attribute this discrepancy to differences in the underlying RoBERTa model and its training methodology across Latin and non-Latin languages, a topic further explored in section 5. To promote reproducibility and facilitate future research endeavors,

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1043–1052

the complete codebase of our project has been shared on GitHub[1].

2 Background

2.1 Dataset Overview

The SemEval-2024 Task 1 is structured into Tracks A, B, and C, each tailored to specific methodologies and objectives. Our focus lies on Track A (Supervised), which utilizes labeled data to train STR systems. The datasets for Task 1 encompass training, development, and test sets across 14 languages, each comprising sentence pairs (Ousidhoum et al., 2024a). Each sentence pair is annotated with a semantic relatedness score, ranging from 0 (indicating no relatedness) to 1 (suggesting strong relatedness). Participants are tasked with predicting the degree of semantic relatedness between sentence pairs, crucial for furthering research in NLP.

2.2 Related Work

The exploration of sentence-level STR has been hindered by the scarcity of available datasets (Abdalla et al., 2021). Existing datasets, such as those compiled by Finkelstein et al. (2002), Gurevych (2006), Panchenko et al. (2016), and Asaadi et al. (2019), predominantly focus on unigram and bigram STR. However, the seminal works of Abdalla et al. (2021), and Ousidhoum et al. (2024a) paved the way for further research by constructing the first sentence-level STR datasets. Traditionally, both STR and STS have been approached using knowledge-based and statistical methods (Sadr, 2020; Chandrasekaran and Mago, 2020). Notable efforts include the application of knowledge bases such as thesauri, ontologies, and dictionaries for STR, as surveyed by Salloum et al. (2020). Statistical methods, on the other hand, leverage features extracted from corpora, with prominent examples including Latent Dirichlet Allocation (LDA) by Blei et al. (2009) and Latent Semantic Analysis (LSA) by Landauer and Dumais (2008) for topic modeling.

In recent years, the application of deep learning methodologies has surpassed traditional approaches in STS tasks. Noteworthy advancements include the Tree-LSTM model proposed by Tai et al. (2015), which outperformed other neural network models in SemEval-2014. He and Lin (2016) introduced a hybrid architecture of Bi-LSTM and CNN, outperforming the Tree-LSTM model on the SICK dataset. Wang et al. (2016) achieved state-of-the-art results using the Word2Vec embeddings model in both the QASent and the WikiQA datasets, while Shao (2017) leveraged GloVe embeddings to achieve the third rank in SemEval-2017.

Several studies have demonstrated that fine-tuning transformer-based models achieves state-of-the-art in comprehending the semantics of textual data. The transformer model, first introduced by Vaswani et al. (2017), employs attention mechanisms to capture word semantics. Later on, Devlin et al. (2019) utilized it to create BERT word embeddings. Subsequently, XLNet, proposed by Yang et al. (2019), surpassed BERT in performance. Consequently, Lan et al. (2019) introduced ALBERT, which outperforms previous models. Additional transformer-based variations of BERT models include TinyBERT (Jiao et al., 2020), RoBERTa (Liu et al., 2019), and DistilBERT (Sanh et al., 2019). Also, Raffel et al. (2019) presented five distinct versions of the T5 transformer model, each varying in parameter size. Their work demonstrated that the performance of these pretrained models improves with larger datasets and enhanced computational resources.

Laskar et al. (2020) addressed sentence similarity modeling within an answer selection task. Through experiments conducted, they showed that fine-tuning RoBERTa model achieves state-of-the-art performance across datasets. Yang et al. (2020) showcased that the RoBERTa-based model achieved superior performance compared to the BERT and XLNET models in a clinical STS task, achieving a Pearson Correlation of 0.90. Similarly, Huang et al. (2021) conducted a comparison of TF-IDF combined with various models including ALBERT, BERT, and RoBERTa for word similarity detection in sentence pairs within Task 2 of SemEval-2021. Their experimental findings substantiated that RoBERTa yielded superior results by 0.846 on the test data. Nasib (2023) addressed reference validation task by employing BERT, SBERT, and RoBERTa. His study illustrated the efficacy of fine-tuning a RoBERTa-based model for text classification tasks, achieving state-of-the-art performance across multiple benchmark datasets. He emphasized that optimizing the model's performance involves activ-

[1] https://github.com/Sharif-SLPL/Sharif-STR

ities such as hyperparameter tuning, regularization, and data augmentation.

Abdalla et al. (2021) conducted an extensive investigation into semantic sentence representation methods, revealing that supervised methods utilizing contextual embeddings, particularly those fine-tuning BERT or RoBERTa, outperform other techniques, reaching a correlation of 0.83. Building upon these findings, we adopt fine-tuning RoBERTa as the primary strategy in this paper. Subsequent sections will detail our system architecture.

3 System Overview

In this section, we present a comprehensive overview of our system's architecture, outlining the key algorithms and modeling decisions that underpin our model.

3.1 Core Algorithms and System Architecture

Our system harnesses the Transformer architecture for its ability to capture long-range dependencies. At its core, we harness the power of a pre-trained RoBERTa model (Liu et al., 2019) for regression analysis, tailoring its parameters to accurately predict a floating-point value from the input text. While RoBERTa isn't explicitly trained for sentence relatedness scoring, its training encompasses an understanding of the relatedness of sentences within discourse, rendering it suitable for our task.

During the pre-training process of RoBERTa, the emphasis is placed on tasks related to NLU. This involves exposing the model to a diverse range of linguistic contexts and training it to comprehend the nuances of language. Our word embeddings utilize an embedding matrix with a dimensionality of 768. Position embeddings and token type embeddings further contribute to the model's comprehension of sequential and contextual information within the input data.

The RobertaEncoder comprises a stack of 12 identical RobertaLayers, each employing a multi-head self-attention mechanism. This mechanism enables the model to concurrently absorb different parts of the input sequence, showing promise in analyzing similarities between various inputs. Following the attention mechanism are intermediate sub-layers and output sub-layers. The intermediate sub-layer employs a fully connected feed-forward network with a GELU activation function, while the output sub-layer is responsible for proper transformation and normalization of features.

The classification head, positioned after the encoder, is tasked with generating the final output for sequence classification. It consists of a linear layer with 768 input features, followed by a dropout layer to prevent over-fitting. An additional linear layer featuring a solitary output neuron enables binary classification. By viewing the problem as a regression task, the classifier yields a linear output designed for a singular class, producing a probabilistic value indicative of the relatedness between input sentences.

3.2 Resources

For training our model, we relied on the dataset provided for SemEval-2024 Task 1 (Ousidhoum et al., 2024a). In addition to the primary dataset, we augmented our training dataset using the T5 model (Raffel et al., 2019). By leveraging T5's paraphrasing capabilities, we explored data augmentation techniques for Track A on the training sets of our dataset but failed to achieve consistent results across experiments. While some experiments showed an increase in model accuracy, in other cases, it did not alter the results. Data augmentation consistently worked well only on the English dataset. More details about data augmentation results and our secondary investigation on Track C are provided in Appendix A and B.

By incorporating both the SemEval-2024 Task 1 dataset (Ousidhoum et al., 2024a) and augmented training data generated by T5, our approach benefits from a comprehensive and diverse set of resources, enabling robust training and evaluation of our STR model across multiple languages and textual domains.

3.3 System Challenges

Augmenting the dataset for training set using T5 paraphrases posed several challenges. Firstly, while the primary dataset was labeled through collaborative human judgment, the augmented data lacked this human validation. This absence of human labeling for the augmented data may potentially impact its quality. Moreover, the augmentation process introduced alterations to the diversity of the data, presenting a challenge to maintaining the original data variety.

The decision to employ data augmentation exclusively for testing purposes raises concerns re-

garding its potential impact on model quality. Addressing these challenges associated with data augmentation is crucial for improving the efficacy of our model. Exploring solutions to mitigate these issues can enhance our approach to tackling the task at hand.

4 Experimental Setup

4.1 Dataset

The dataset statistics utilized for each language are presented in Table 1:

As shown in Table 1, approximately 0.8 of the Task 1 dataset is allocated for system training, while the remainder is reserved for evaluation. The limited availability of training data necessitates cautious consideration during testing, as the model's performance may be influenced by the scarcity of training instances. Additionally, the entire development set is utilized for model selection.

4.2 Pre-processing and Hyper-Parameter Tuning

A crucial aspect of our pre-processing involves converting the labels (scores) of each data instance to float values, ensuring compatibility with the model's expected input format. Furthermore, the input texts undergo tokenization using the RoBERTa-tokenizer both during training and inference.

Hyperparameter tuning plays a pivotal role in optimizing model performance. Our tuning process encompasses exploring various hyper-parameters, including learning rates in the range of [0.00001, 0.00003], dropout rates ranging from [0.1, 0.3], batch sizes spanning [4, 32], and token sizes from [32, 128]. Through iterative experimentation, we determined that a learning rate of 0.00003, a dropout rate of 0.1, a token size of 128, a batch size of 16, and a weight decay of 0.01 yield optimal results across all languages.

The selection of an appropriate token size is not solely based on computational considerations; rather, it is informed by dataset analysis. Upon examination, it became evident that the majority of data instances are predominantly short, aligning with our token size choice. Additionally, truncation during tokenization supports the chosen token size, ensuring efficient model training without sacrificing data representativeness.

4.2.1 Mean Squared Error (MSE)

Mean Squared Error quantifies the average of the squared differences between predicted and actual values. It is calculated using the formula:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (1)$$

Where N is the number of instances, y_i is the true label, and $\hat{y}_i$ is the predicted value.Additionally, Mean Absolute Error computes the average absolute differences between predicted and actual values.Moreover, the R-squared score assesses the proportion of variance in the dependent variable explained by the independent variable.

These evaluation measures collectively shed light on our regression model's performance in predicting the degree of relatedness between text samples. Using these metrics together enables the monitoring of the model's performance and, hence, facilitates decisions on hyper-parameters, model selection, etc. The evaluation method and hyper-parameter choices remain consistent across all models and languages. For the analysis of results presented in Section 5, the obtained scores were discretized and categorized into five distinct ranges to enhance visual understanding.

5 Results

5.1 Findings

A direct comparison with previous models and datasets similar to this task is challenging due to our specific focus on fine-tuning the RoBERTa model and utilizing the dataset provided by Ousidhoum et al. (2024a). Drawing from the insights of Raffel et al. (2019) working on the STS dataset, it is evident that the performance of transformer models improves with larger training corpora and enhanced computational resources. Raffel et al. (2019) demonstrated that the RoBERTa transformer-based model achieved a Pearson correlation of 0.922, surpassing ERNIE 2.0, DistilBERT, and TinyBERT on STS dataset benchmarks. Conversely, ALBERT, XLNet, and T5-11B outperformed RoBERTa on the same task, achieving a Pearson correlation of 0.925. Therefore, we recommend conducting a benchmark study of top-performing transformer models like RoBERTa, ALBERT, XLNet, and T5-11B in future research endeavors. Using the official metric of Spearman Correlation proposed in SemEval-2024 Task 1 (Ousidhoum et al., 2024b), our system achieves the following scores on different data splits and languages:

As shown in Table 2, Firstly, comparing the performance between English, Spanish, and Arabic

Language/Split	Dataset	Train	Testset	Devset
English	5752	4400	1101	251
Spanish	1702	1249	313	140
Arabic	1360	1009	252	97

Table 1: Dataset Statistics

Language/Split	Devset	Testset(Competition)
English	0.83	0.82
Spanish	0.71	0.67
Arabic	0.32	0.38

Table 2: Correlation Metric Scores

models, we observe varying degrees of success. The English model demonstrates the highest Spearman Correlation scores, both on the development and test sets, with scores of 0.83 and 0.82, respectively. This indicates that the English model performs relatively well in capturing the semantic relatedness between text pairs. Similarly, the Spanish model also achieves respectable scores, albeit slightly lower, with scores of 0.71 on the development set and 0.67 on the test set. However, the Arabic model lags significantly behind, exhibiting notably lower scores of 0.32 on the development set and 0.38 on the test set.

The disparity in performance between the Arabic model and the English and Spanish models could be attributed to several factors. One possible explanation is the availability and quality of training data. The Arabic dataset may suffer from a scarcity of labeled instances, resulting in a less robust model. Additionally, linguistic and structural differences between Arabic and Latin languages may pose challenges for the model in accurately capturing semantic relatedness. This discrepancy underscores the importance of adequately addressing language-specific characteristics and challenges in model development.

Furthermore, the analysis of the Arabic model's performance on the test set reveals a noteworthy observation. Despite achieving a relatively low Spearman Correlation score, the model appears to disproportionately classify most inputs as highly related. This discrepancy suggests a potential limitation in the model's ability to discern varying degrees of relatedness accurately. It implies that while the model may perform adequately in certain aspects, such as overall correlation with human annotations, it may struggle with nuanced interpretations of relatedness levels in real-world scenarios. The output of the model is provided in Appendix D.

The scatter plots depicted in Figure 1, respectively for English, Spanish, and Arabic, illustrate the correlation between the model predictions and human annotations. The English model closely aligns with human annotations, while the Spanish model exhibits an even closer alignment on certain inputs. However, the Arabic model's performance varies, indicating discrepancies between predicted and actual relatedness scores. These findings underscore the importance of dataset size and linguistic nuances in model performance across different languages. Further investigation is warranted to elucidate the factors influencing model behavior and to improve performance, particularly in languages with limited training data.

5.2 Error Analysis

While confusion matrices are less commonly utilized in regression problems, discretizing the model's scores allows us to glean insights into its performance. Confusion matrix plots for English, Spanish, and Arabic are provided in Figure 2, respectively. Upon examining the confusion matrix of the English dataset, it becomes apparent that the model performs well within certain score ranges. However, there are notable areas, particularly within the highly related range (0.6-1.0), where our model could benefit from improvement.

A similar observation holds true for the Spanish dataset, where the model demonstrates proficiency in predicting less related sentences but encounters challenges with highly related ones. Conversely, the Arabic dataset presents a markedly different scenario. While the majority of predictions fall within the mid-range of relatedness, they are predominantly incorrect.

Based on the histogram and extracted statistics from the fine-tuning data in Figure 3 in Appendix C, it appears that the majority of the training data has a distribution centered around the median (Spanish

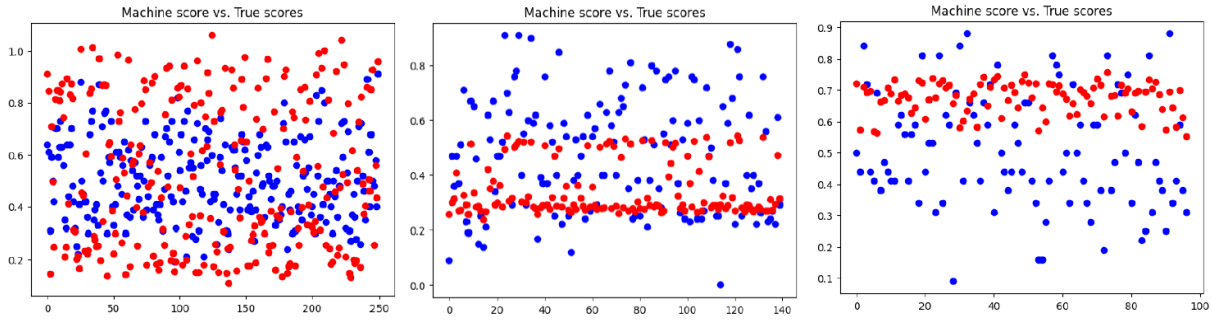


Figure 1: Scatter Plots of English, Arabic and Spanish Languages

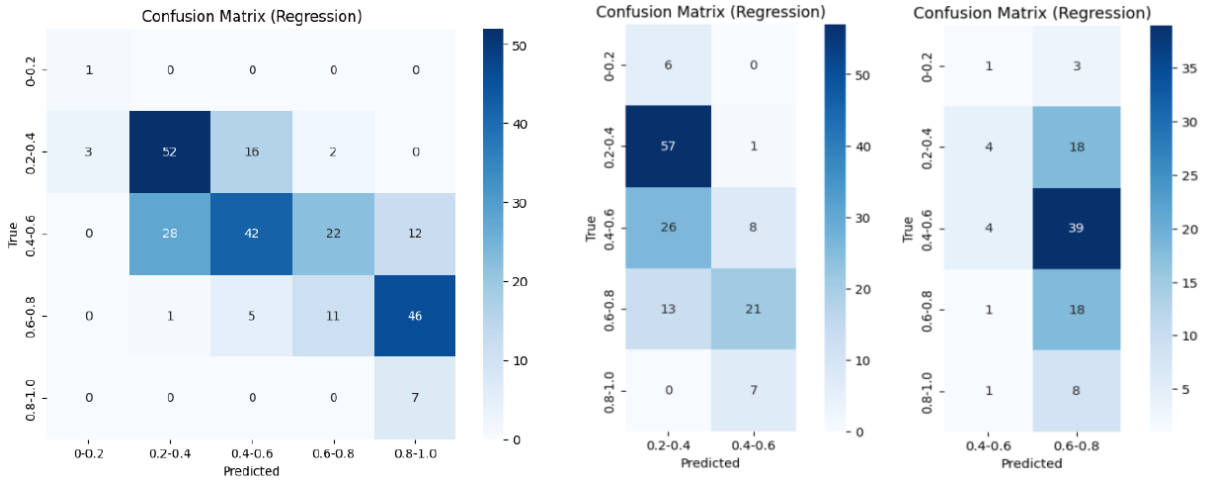


Figure 2: The Confusion Matrix Plot of English, Arabic and Spanish Languages

Mean Score: 0.43, Arabic Mean Score: 0.50). Consequently, finte-tuned Arabic and Spanish models seem to have less capability in understanding data on both ends of the spectrum.

These insights highlight the model's strengths and weaknesses across different datasets and underscore the need for further investigation into improving performance, particularly in accurately predicting highly related sentences across all languages. Further exploration of the factors contributing to model errors, such as dataset characteristics and linguistic nuances, is essential for refining the model's predictive capabilities.

6 Conclusion

In our investigation, we focused on fine-tuning RoBERTa for STR, primarily targeting Latin languages like English(0.82) and Spanish(0.67). While our approach showed promising results for these languages, particularly in achieving high correlation, the outlook was less favorable for Arabic(0.38). This echoes discussions in previous works, emphasizing the significant influence of the data on model performance. Our exploration into Track C, which is given in Appendix B, further enriched our understanding of the challenges and opportunities in STR system development. As a contribution to the field, we put forth several recommendations for enhancing STR systems. Firstly, we propose the development of additional Transformer models trained on diverse language families, focusing on languages that share similarities with Latin languages. Furthermore, a comprehensive benchmark of models on the STR dataset is essential, building on previous research that highlights the strong performance of models like ALBERT, XLNet, and T5-11B on the STS dataset. Moreover, the utilization of translation techniques and data augmentation methods could enhance model performance, particularly for languages with limited training data. In conclusion, our study sheds light on the nuances of STR system development and underscores the importance of considering language-specific factors and domain characteristics. By pursuing the avenues outlined in this paper, we aim to contribute to the advancement of STR research and facilitate the development of more robust and accurate models for NLU tasks.

Acknowledgments

We express our gratitude to the Speech and Language Processing Laboratory at Sharif University of Technology[2] for offering us the opportunity for collaborative work.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2021. What makes sentences semantically related? a textual relatedness dataset and empirical study. *ArXiv*, abs/2110.04845.

Shima Asaadi, Saif M. Mohammad, and Svetlana Kiritchenko. 2019. Big bird: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In *North American Chapter of the Association for Computational Linguistics*.

David M. Blei, A. Ng, and Michael I. Jordan. 2009. Latent dirichlet allocation.

Dhivya Chandrasekaran and Vijay Mago. 2020. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54:1 – 37.

Fuzan Chen, Chenghua Lu, Harris Wu, and Minqiang Li. 2017. A semantic similarity measure integrating multiple conceptual relationships for web service discovery. *Expert Syst. Appl.*, 67:19–31.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.*, 20:116–131.

Marc Franco-Salvador, Paolo Rosso, and Manuel Montes y Gómez. 2016. A systematic study of knowledge graph analysis for cross-language plagiarism detection. *Inf. Process. Manag.*, 52:550–570.

Iryna Gurevych. 2006. Thinking beyond the nouns - computing semantic relatedness across parts of speech.

Hua He and Jimmy J. Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *North American Chapter of the Association for Computational Linguistics*.

Bo Huang, Yang Bai, and Xiaobing Zhou. 2021. hub at semeval-2021 task 2: Word meaning similarity prediction model based on roberta and word frequency. In *International Workshop on Semantic Evaluation*.

[2] https://github.com/Sharif-SLPL

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2009. *Speech and language processing*, 2. ed., [pearson international edition] edition. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, London [u.a.].

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Thomas K. Landauer and Susan T. Dumais. 2008. Latent semantic analysis. *Scholarpedia*, 3:4356.

Md Tahmid Rahman Laskar, Xiangji Huang, and Enamul Hoque. 2020. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *International Conference on Language Resources and Evaluation*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Abdullah Umar Nasib. 2023. References validation in scholarly articles using roberta. Project report, Brac University.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *Preprint*, arXiv:2402.08638.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Alexander Panchenko, Dmitry Ustalov, Nikolay Arefyev, Denis Paperno, Natalia Konstantinova, Natalia V. Loukachevitch, and Chris Biemann. 2016. Human and machine judgements for russian semantic relatedness. *ArXiv*, abs/1708.09702.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Hossein Sadr. 2020. Exploring the efficiency of topic-based models in computing semantic relatedness of geographic terms.

Said A. Salloum, Rehan Khan, and Khaled F. Shaalan. 2020. A survey of semantic analysis approaches. In *International Conferences on Artificial Intelligence and Computer Vision*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Yang Shao. 2017. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *International Workshop on Semantic Evaluation*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *ArXiv*, abs/1503.00075.

Mohamed Ali Hadj Taieb, Torsten Zesch, and Mohamed Ben Aouicha. 2019. A survey of semantic relatedness evaluation datasets and procedures. *Artificial Intelligence Review*, 53:4407 – 4448.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *International Conference on Computational Linguistics*.

Xi Yang, Xing He, Hansi Zhang, Yinghan Ma, Jiang Bian, and Yonghui Wu. 2020. Measurement of semantic textual similarity in clinical texts: Comparison of transformer-based models. *JMIR Medical Informatics*, 8.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*.

A Data Augmentation Results

As we describe data augmentation in section 3.2, we use T5 model to augment some training data and use them in training of model.So in this section we show results of data augmentation effect on Pearson Correlation for English language in Table 3.

Model hyper parameters		without data augmentation	with data augmentation
Learning rate	3e-5	0.79	0.81
Max length	128		
Batch size	16		
Epoch	4		

Table 3: Data Augmentation Affect on Pearson Correlation

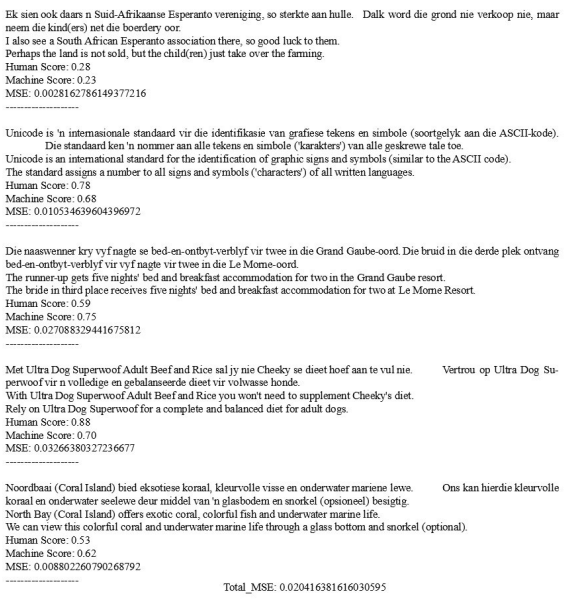


Figure 3: Output of Afrikaans

B Track C - Cross-Lingual

Using the translation method in Track C, we employed our Track A model trained on English language. The input sentences were first translated into English using the Google Translate API, followed by the utilization of the trained Track A model. The evaluation results demonstrate promising performance across some languages with this approach. However, errors might arise from either the Google Translate API or the model itself. Exploring alternative translation APIs could potentially enhance the overall performance. Figures 3, 4, and 5 display the outputs in Afrikaans, Amharic, and Modern Standard Arabic. Additionally, the high-quality output images are provided in our GitHub project.

Test Data	Pearson Correlation	MSE
afr_test_with_labels.csv	0.8	0.0204
amh_test_with_labels.csv	0.73	0.0309
arb_test_with_labels.csv	0.51	0.0431

Table 4: Track C Results

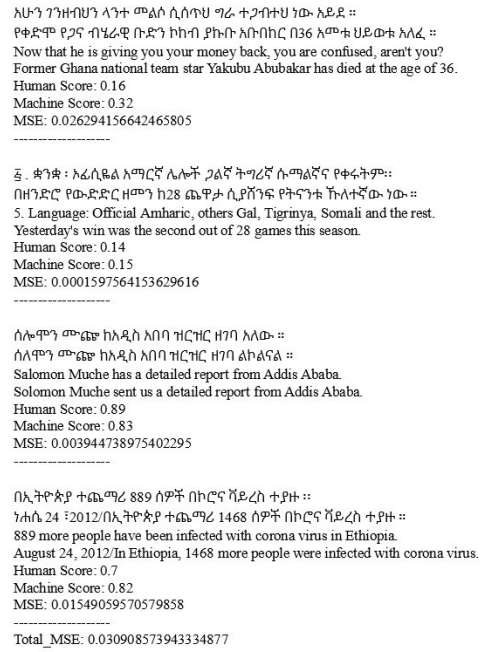


Figure 4: Output of Amharic Language

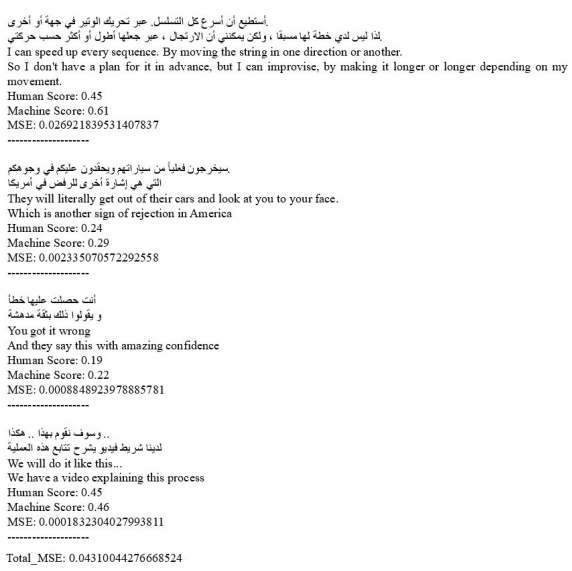


Figure 5: Output of Modern Standard Arabic

C Histogram of Spanish and Arabic Languages

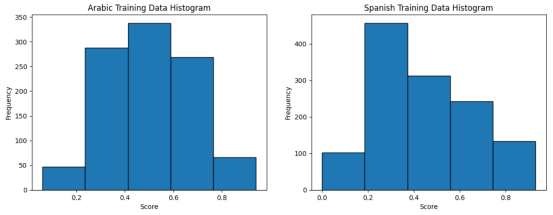


Figure 6: Histogram of Spanish and Arabic Training Dataset

D Outputs of Track A (Supervised)

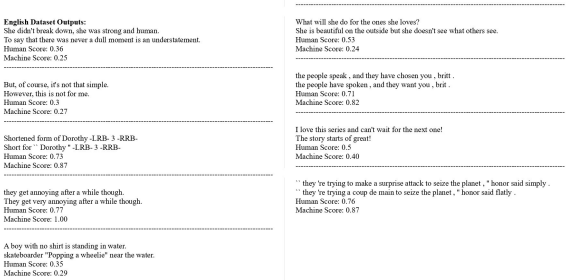


Figure 7: Output of English Language

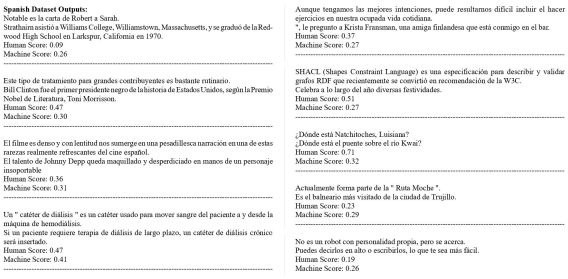


Figure 8: Output of Spanish Language

Figure 9: Output of Arabic Language

DUTh at SemEval 2024 Task 5: A multi-task learning approach for the Legal Argument Reasoning Task in Civil Procedure

Ioannis Maslaris **Avi Arampatzis**

Database & Information Retrieval research unit,
Department of Electrical & Computer Engineering,
Democritus University of Thrace, Greece.
{imaslari,avi}@ee.duth.gr

Abstract

Text-generative models have proven to be good reasoners. Although reasoning abilities are mostly observed in larger language models, a number of strategies try to transfer this skill to smaller language models. This paper presents our approach to SemEval 2024 Task-5: The Legal Argument Reasoning Task in Civil Procedure. This shared task aims to develop a system that efficiently handles a multiple-choice question-answering task in the context of the US civil procedure domain. The dataset provides a human-generated rationale for each answer. Given the complexity of legal issues, this task certainly challenges the reasoning abilities of LLMs and AI systems in general. Our work explores fine-tuning an LLM as a correct/incorrect answer classifier. In this context, we are making use of multi-task learning to incorporate the rationales into the fine-tuning process.

1 Introduction

In recent years, Large Language Models (LLM) development has witnessed unprecedented advancements, with Large Language Models such as GPT-3 demonstrating remarkable capabilities in understanding and generating human-like text. However, the effectiveness of these models in reasoning tasks remains an area of ongoing exploration and enhancement. While LLMs excel in linguistic fluency and context understanding, their capacity for reasoning often falls short of human-level comprehension (Huang and Chang, 2022).

In this paper, we describe the DUTh participation in *SemEval 2024 Task 5: The Legal Argument Reasoning Task in Civil Procedure* (Bongard et al., 2022)[1], on leveraging the reasoning capabilities of Large Language Models for multiple-choice question-answering in the context of US civil procedure. The task can be formulated as follows: *given an introduction to a case, a question, and a candidate answer, classify if the given answer is correct or wrong*. The dataset is based on *The Glannon Guide To Civil Procedure* by Joseph Glannons (Glannon, 2023). The multiple-choice questions come from the book's exercises, which aim to test the reader.

The training set has a size of 666 entries, which is smaller compared to other similar datasets aiming to examine the capabilities of LLMs using human-generated rationales (Hancock et al., 2019). Although the complexity of legal domain text and the number of details engulfed in real legal cases are large. The cognitive skills required to understand and handle legal cases make this task an interesting challenge for LLMs' reasoning abilities.

Our proposed system is a LegalBERT (Chalkidis et al., 2020) classifier, fine-tuned on a downstream task incorporating rationales for each answer. Our code implementation builds on the organizers' and is publicly available.[2] Additionally, we experimented with a multi-task Flan-T5 model. This strategy involves a different way to use rationales in the training process. The model is trained to predict the correct labels and, at the same time, generate relevant rationales. Its performance is evaluated through a custom loss function that accounts for the loss of the label prediction task and the loss of the rationale generation task separately. Although it did not surpass the performance of the LegalBERT classifier, it is an interesting approach that can be further examined on the current task.

2 Background

2.1 Related Work

Large Language Models have demonstrated remarkable few-shot capabilities (Smith et al., 2022; Zhang et al., 2022). This models, having more than 100 billion parameters, prove to be difficult to be

[1] https://codalab.lisn.upsaclay.fr/competitions/14817

[2] https://github.com/DataMas/SemEval2024-Task5

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1053–1057

deployed for regular real-world applications. For this reason, a lot of effort is being made toward leveraging the reasoning capabilities of smaller language models.

Knowledge distillation is a fine-tuning strategy aiming to transfer knowledge from larger and more complex models into smaller and more practical models. The larger teacher model acts on a dataset to predict its labels, and then the smaller student model is trained on these generated labels. In fact, distillation can be performed on unlabeled or limited labeled data (Abbasi et al., 2021; Fu et al., 2023).

Based on this idea, rationales can also be used to supervise the fine-tuning process of a smaller model. Human-generated rationales have been used as auxiliary inputs to improve the model's performance (Fatema Rajani et al., 2019). Another approach involves using these rationales as labels in order to make a model generate similar explanations for its predictions(Eisenstein et al., 2022). Learning from LLM-generated rationales is a relatively new field of experimentation. Larger Language Models can explain their predictions by generating reasoning steps (Kojima et al., 2022). This reasoning steps can be used in the same way as human-generated rationales to improve the performance of smaller models (Pruthi et al., 2022).

Taking the previous ideas one step further, (Hsieh et al., 2023) proposed a multi-task fine-tuning framework. They essentially train a model on two separate tasks at the same time. The model is trained to not only predict the correct labels but also to generate accurate rationales explaining its predictions. They extract rationales from LLMs using Chain-of-Thought prompting. With their multi-task training, they are able to fine-tune smaller language models, which perform comparable to or better than larger models. They achieve not only to reduce the size of the final models but also the size of the needed data.

2.2 Dataset

The organizers provide a dataset from the US legal domain in English. It is essentially a multiple-choice question-answer dataset. It contains questions and possible answers regarding topics of US civil procedure. Every question concerns a legal case. Along with each question, a paragraph serving as a general introduction to the case is provided. Every possible answer is accompanied by an analysis of why its context is relevant to the case. Additionally, for every batch of possible answers corresponding to a question, a paragraph with general comments discussing all answers' rationales is given.

The training and development sets are compiled of all the features discussed above (introduction, question, answer, analysis, and explanation), while the test set excludes the features giving reasoning behind every answer. The train, development, and test sets consist of 666, 84, and 98 entries, respectively. Following, we can see the structure of the dataset clearly. The items without bold annotations are not included in the test set.

- **"introduction":** A paragraph regarding the context of the question.
- **"question":** The question regarding a legal case.
- **"answer":** A possible answer to the question.
- **"label":** A binary indicator for correct and wrong answer.
- "analysis": Reasoning on why each answer is right or wrong.
- "explanation": A paragraph discusing the reasoning of all possible answers to a question.

2.3 Evaluation Measures

Submissions are evaluated by two metrics:

- F1 score: The F1 score is defined as the harmonic mean of precision and recall, offering a single metric to asses a classifier's performance by considering both false positives and false negatives.
- Accuracy: Accuracy score is a measure used to evaluate the performance of a classification model. It is defined as the ratio of correctly predicted observations to the total observations.

Finally, participants are ranked based on the F1 score of their system.

3 System Overview

3.1 Data Pre-processing

Pre-trained Large Language Models are constrained by the maximum length of text input they can process. This limitation arises from the model's fixed-sized input layer, which can only accommodate a certain number of tokens (e.g., words or sub-words). We want to use the *introduction* and

analysis as context for the *question* and *answer* accordingly. This makes the final input exceed its token limit. The models we have used for our experiments have a limit of 512 tokens. Combining the *introduction, question, analysis* and *answer* creates input instances of length greater than 700 words on average.

In order to fit the constructed input instances, we have employed a sliding window mechanism. We use the same Sliding Window Complex (SWC) strategy proposed by the organizers (Bongard et al., 2022)[3]. This sliding window algorithm splits the inputs into chunks of specified length L, which is smaller than the limit length. In order for everything to fit, some features must be sliced. The specific details on how the features are sliced will be described later in the System architecture and Multi-task learning sub-sections. For example, one approach is to concatenate *explanation* and *answer* by keeping the whole *answer* to every chunk and pad the explanation until the limit of words is reached.

3.2 System Architecture

For our system, we utilize Legal-BERT to classify every chunk as wrong or correct. We also evaluated BERT (Devlin et al., 2018) and DistilBERT (Sanh et al., 2019), but they proved inferior. We fine-tune each model with constructed input instances. These instances are compiled by the *question*, *introduction*, *answer*, and *analysis*. The way these features are concatenated to form an input is to keep the whole *question*, *analysis*, and *answer*. The rest of the available space is filled with a part of the *introduction*.

Before the *question*, we add the distinctive feature *Q:*, while we also do the same thing before the *answer* with the distinctive feature *A:*. This way, we are making it clear to the model when a question and an answer begin. We have experimented with the learning rate and weight decay and used Optuna on the best performing model for hyperparameter optimization. Finally, training was terminated with early-stopping, with patience being set to 10.

3.3 Multi-task Learning

Based on the idea of *Distilling-step-by-step* (Hsieh et al., 2023)[4], we implement a similar system where we use the *analysis* feature provided in the dataset as rationale. During the training process, the model is trained to both predict the correct label and, at the same time, generate a comprehensive rationale for every input instance. This is done through the use of a custom loss function that accounts for the label prediction error and the rationale generation error.

$$L = (1 - w)L_{\text{label}} + wL_{\text{rationale}}$$

Adding the rationale generation loss to the training process helps the model better understand the logic behind why every answer is correct or wrong. The loss function is weighted with a factor w. Through w, we can control which task the model should focus on more during training. Choosing a $w = 0.5$ means that the model will try equally to learn both tasks. For values $w < 0.5$ the model places more importance on learning to predict labels correctly, and for values $w > 0.5$ the model is more focused on learning to generate accurate rationales.

We use the sliding window to create the input instances. These consist of an *introduction*, a *question*, and an *answer*. The distinctive features *Q:* and *A:* are also used here in the same way as described in the previous paragraph. In order to fit every instance into the limit of input tokens, every chunk has the complete *question* and *answer* and is padded with part of the *introduction*. In order to help the model distinguish between the two tasks, every instance is padded with another distinctive feature. For the label prediction task, we use the feature *Predict:* at the beginning of the instance. For the rationale generation task, we use the feature *Explain:*. This is done on our custom data collator function[5] and the result is two separate datasets.

In order to train the model in a multi-task manner, we created a custom trainer function. In this function, the model is prompted separately with the two task-specific datasets coming from the data collator. The answers of the model are evaluated, and the loss is computed through the custom function we described earlier. Finally, we define the prediction step of the model to produce answers for both tasks.

For this strategy, we utilize the small version of the Flan-T5 model (Chung et al., 2022). Because the model is trained to generate rationales, it must receive the labels as text. We transformed the labels of 0 to *Wrong* and the labels of 1 to *Correct*. Consequently, when the model is prompted to predict the

[3] github.com/trusthlt/legal-argument-reasoning-task
[4] github.com/google-research/distilling-step-by-step
[5] huggingface.co/docs/transformers/main_classes

labels of new data, it will respond with *Correct* or *Wrong*. For this reason, we have to convert the text responses to 1–0 accordingly in order to evaluate them and submit our results.

The final multi-task trained system receives a dataset and first preprocesses it. It concatenates the *introduction*, *question*, and *answer* and converts the labels to *Correct* - *Wrong* text. The model is prompted with the instances, and its responses are converted into 1–0.

Pre-trained model	F1-score	Accuracy
LegalBERT	0.5382	0.6837
BERT	0.5081	0.7245
DistilBERT	0.4269	0.7245
LegalBERT*	0.4827	0.7245
Multi-task Flan-T5	0.5324	0.6224

Table 1: Best performance of each model. *This is the score the best-performing model achieved during the evaluation phase. All the other scores have been achieved during the post-evaluation phase and are not counted for the leaderboard.

4 Experimental Setup

4.1 Chain of Thought

Before creating embeddings, we tried to fine-tune the models using a Chain of Thought strategy. During the sliding window process, we used auxiliary phrases to make the final input make more sense to the model. For example, we used the phrase *Based on the following* before adding the part of the *introduction*. After the *introduction* and before the question, we added the phrase *Answer the following question*. For the answer-analysis part, we used the phrases *The following answer* followed by the answer and *is correct/wrong because* followed by the analysis.

Although a widely used and promising technique, CoT did not prove to increase the performance of our models. At least based on the phrases and the arrangement we used. The task prefixes *Predict* and *Explain* that we used for the multi-task system can also be considered as a CoT approach. On this occasion, they were efficient in guiding the model to distinguish between the two tasks.

4.2 Experiments

Our experiments are mainly focused on fine-tuning different models under different hyperparameters. The hyperparameters we experimented on were the learning rate and the weight decay. We We came up with the best set of hyperparameters through optimization using the Optuna hyperparameter optimization framework. [6] In the first set of experiments regarding fine-tuning on a downstream classification task, we evaluated three pre-trained models: BERT, LegalBERT, and DistilBERT. The best-performing model proved to be the LegalBERT. For the second set of experiments regarding multi-task fine-tuning, we utilized the small version of the Flan-T5 model. The same hyperparameter optimization procedure was followed. We also experimented with the parameter w which controls the amount of focus on each task. A weight $w = 0.5$ proved to be slightly better.

5 Results

The comprehensive scores of our systems across the utilised models are presented on Table 1. The highest F1 score was 0.5324 achieved by LegalBERT, followed closely by the multi-task T5. According to accuracy, BERT and DistilBERT perform better with a score of 0.7245, and LegalBERT comes in second with 0.6837. LegalBERT . Although our models do not perform well, we can make some assumptions on why that is.

Firstly, regarding LegalBERT, it is possible that simply adding the rationale to the input along with the *introduction*, *question* and *answer* will not helping the model learn the logic behind justifying each answer. In fact, it makes the model perform worse compared to setups where only *introduction*, *question* and *answer* is used (Bongard et al., 2022). Additionally, our multi-task system, although incorporating a more complex training mechanism, it does not seem to be able to distinguish answers efficiently. The small version of Flan T5 is only of 80 million parameters. At this scale, it might be difficult for language models to grasp complex concepts laying on rationales. This, in fact, can be confirmed by prompting the multi-task model to generate rationales based on the input. The generated rationales barely makes any sense.

6 Conclusion

Through our experiments, we could not find a significantly performing system. Even the multi-task approach, which makes good use of the rationales to better establish a connection between input and

[6] https://github.com/optuna/optuna

label, could not perform well. But we demonstrated the possible limitations and difficulties of such tasks, where logical reasoning is needed in order for a model to perform well.

The primary benefit of multi-task learning lies in the use of rationales, enabling the model to perceive the reasons behind the correctness or incorrectness of every answer. In this work, our capabilities were constrained by hardware limitations, leading us to experiment with a smaller Language Model. However, this model's capacity to comprehend longer content is limited by its size.

Next steps could involve experimentation with bigger Language Models regarding the multi-task approach. We believe that a larger model could better grasp the context of the rationales and draw better associations between a question and possible answers. Another approach regarding the multi-task strategy is to incorporate rationales through a more efficient loss function. Another weighing strategy could be used, for example.

References

Sajjad Abbasi, Mohsen Hajabdollahi, Pejman Khadivi, Nader Karimi, Roshanak Roshandel, Shahram Shirani, and Shadrokh Samavi. 2021. Classification of diabetic retinopathy using unlabeled data and knowledge distillation. *Artificial Intelligence in Medicine*, 121:102176.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. *arXiv preprint arXiv:2211.02950*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Eisenstein, Daniel Andor, Bernd Bohnet, Michael Collins, and David Mimno. 2022. Honest students from untrusted teachers: Learning an interpretable question-answering pipeline from a pretrained language model. *arXiv preprint arXiv:2210.02498*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv e-prints*, pages arXiv–1906.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. *arXiv preprint arXiv:2301.12726*.

Joseph W Glannon. 2023. *Glannon guide to civil procedure: learning civil procedure through multiple-choice questions and analysis*. Aspen Publishing.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C Lipton, Graham Neubig, and William W Cohen. 2022. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

MAMET at SemEval-2024 Task 7: Supervised Enhanced Reasoning Agent Model

Mahmood Kalantari
Iran University of Science and Technology (IUST)
m_kalantari76@comp.iust.ac.ir

Mehdi Feghhi
Iran University of Science and Technology (IUST)
feghhi_me@comp.iust.ac.ir

Taha Khany Alamooti
Iran University of Science and Technology (IUST)
khany_taha@comp.iust.ac.ir

Abstract

In the intersection of language understanding and numerical reasoning, a formidable challenge arises in natural language processing (NLP). Our study delves into the realm of NumEval, focusing on numeral-aware language understanding and generation using the QP, QQA and QNLI datasets[1]. We harness the potential of the Orca2 model, Fine-tuning it in both normal and Chain-of-Thought modes with prompt tuning to enhance accuracy. Despite initial conjectures, our findings reveal intriguing disparities in model performance. While standard training methodologies yield commendable accuracy rates. The core contribution of this work lies in its elucidation of the intricate interplay between dataset sequencing and model performance. We expected to achieve a general model with the Fine Tuning model on the QP and QNLI datasets respectively, which has good accuracy in all three datasets. However, this goal was not achieved, and in order to achieve this goal, we introduce our structure 1.

1 Introduction

In the realm of natural language understanding (NLU), the quest for models capable of comprehending and reasoning with textual data has been a longstanding pursuit. The NumEval task, focusing on Numeral-Aware Language Understanding and Generation, stands at the frontier of this endeavor, challenging researchers to develop models adept at grasping numerical information embedded within linguistic contexts. In this study, we delve into the intricacies of fine-tuning methodologies and their impact on the performance of language models, particularly focusing on the QP, QQA and QNLI datasets. (num)

The primary challenge in NLU lies in imbuing models with the ability to interpret and reason with textual information akin to human cognition. Traditional approaches often face hurdles in capturing the nuances of language, especially when numerical data intertwines with linguistic expressions. One possible cause of this problem is that numerals can have various notations, some of which are difficult to understand from their subwords. While models like Orca2 (Mitra et al., 2023), an instance of Large Language Models (LLMs), exhibit remarkable capabilities, their performance nuances in understanding numeral-aware contexts warrant deeper exploration.

The QP, QQA and QNLI datasets(Chen et al., 2023a), (Chen et al., 2019), (Ravichander et al., 2019), (Mishra et al., 2022) serve as test for evaluating the efficacy of language models in understanding questions, question-answering and natural language inference, respectively. These datasets present a diverse array of linguistic challenges, including numeral-aware reasoning, prompting the need for sophisticated training strategies.

Our study uses the Orca2 model, an advanced LLM known for its language comprehension skills. Through meticulous fine-tuning and evaluation on the QP, QQA and QNLI datasets.

We aim to catalyze discourse and innovation in the field of NLU, steering towards more robust and nuanced language models capable of navigating the complexities of numeral-aware language understanding and generation in real-world scenarios.

However, the road to achieving robust language understanding is fraught with challenges, chief among them being the inherent ambiguity and variability present in natural language. Numerical information adds an additional layer of complexity, requiring models to not only parse linguistic constructs but also interpret and reason with numerical data embedded within textual contexts.

Conventional training methodologies, while effective to a certain extent, often fall short in encapsulating the intricate interplay between linguistic semantics and numerical reasoning. The advent of large-scale language models has undoubtedly propelled the field forward, but their performance on numeral-aware tasks remains an area ripe for exploration and refinement.

After conducting several experiments, we have determined that fine-tuning a model for a specific subtask yields significantly higher accuracy compared to fine-tuning a model across all subtasks. Our attempt to fine-tune a generalized model across all subtasks while maintaining accuracy proved unsuccessful. Upon reviewing the results, we recognized the effectiveness of the Orca 2 model utilizing the LORA method for each subtask. Consequently, we trained the model using QLORA, resulting in improved accuracy. To establish a robust framework for addressing a range of reasoning

[1] https://drive.google.com/drive/folders/1mKbiL420U4Ih-hGmpaSki0FCvGHH3Au2?usp=sharing

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1058–1063

subtasks, we propose a structured approach that employs an agent as a supervisor capable of categorizing subtasks. This agent determines which of our fine-tuned models should address each task.

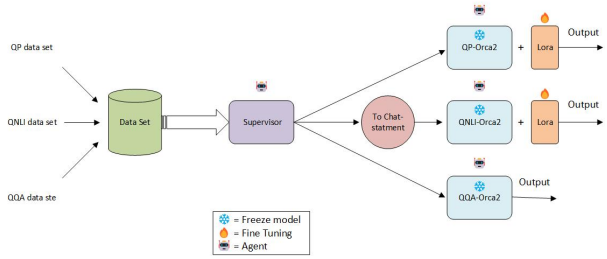


Figure 1: Supervised Enhanced Reasoning Agent Model.

2 Related Work

The intersection of Numerical Evaluation (NumEval) and Natural Language Processing (NLP) has witnessed a surge in research endeavors aimed at refining models' numerical comprehension and processing capabilities. Within this burgeoning field, a multitude of studies have delved into diverse methodologies and frameworks to deepen our understanding of numeracy in linguistic contexts.(Chen et al., 2023b)

A seminal study by (Chen et al., 2021) introduced the utilization of digit-based encoders to represent numerals, laying the groundwork for subsequent investigations into numerical representation methods. Expanding upon this, (Zhang et al., 2020) pioneered the exploration of scientific notation for numerical representation, shedding light on the efficacy of alternative numerical formats in quantitative skill tasks. These methodologies not only deepen our understanding of numerals' linguistic representation but also pave the way for novel approaches in numerical evaluation tasks.

Pretraining tasks have played a pivotal role in enhancing language models' capabilities in comprehending numerical language. (Devlin et al., 2019) revolutionized the field by introducing masked language model (MLM) and next sentence prediction (NSP) tasks, ushering in a new era of transformer-based NLP research. Building upon this foundation, (Yasunaga et al., 2022) proposed the document relation prediction (DRP) task, specifically designed to bolster models' performance in multi-hop reasoning and multi-document understanding tasks. These pretraining paradigms have significantly enriched models' numerical understanding capabilities, underscoring the pivotal role of pretraining in enhancing NLP models' numerical acumen.

In parallel, pre-finetuning strategies have emerged as a promising avenue for enhancing models' numerical comprehension. The Comparative Numbers Dataset (CND), introduced as a pre-finetuning resource, has garnered attention for its efficacy in enhancing models' numerical reasoning abilities. Experimental investigations leveraging BERT, RoBERTa, LinkBERT and FinBERT (Araci, 2019) have demonstrated notable improvements in models' performance across various numerical evaluation tasks, underscoring the potential of pre-finetuning methodologies in augmenting models' numerical understanding.(Chen et al., 2023b)

Generally, the landscape of NumEval research is characterized by a dynamic interplay of numerical representation methods, pretraining tasks and pre-finetuning strategies, each contributing to the advancement of language models' proficiency in numerical understanding and processing.(Chen et al., 2023b)

The results of our tests are very promising in this field and on the tested datasets, the accuracy is higher than the accuracy of the reference article.

3 Approach

Through our exploration with the Fine Tuning model, we discovered commendable accuracy in each sub-task individually. Upon scrutinizing the test outcomes, a noteworthy observation emerged: encoding numerical values within the text as statement-char significantly boosts accuracy in the QNLI task. However, a pertinent challenge persists: determining the appropriate agent for input assignment.

To address this challenge, we devised a framework. Initially, the input undergoes classification by an agent, ensuring its allocation to the most suitable model. In the case of the QNLI task, we preprocess words into statement-char format before directing them to the designated agent.

Furthermore, leveraging the Orca model's remarkable 100% accuracy in the QQA task, we opted to employ the base model as our agent. This strategic decision underscores our commitment to optimizing task performance and model efficacy.

3.1 Baseline Model:

The Orca2 model, a variant of the large language model (LLM), served as the cornerstone of our experiments. Built upon state-of-the-art architecture, Orca2 harnesses the power of deep neural networks to comprehend and generate human-like text responses. Leveraging its pre-trained weights, we fine-tuned Orca2 on the task-specific datasets to imbue it with numeral-aware capabilities.

3.2 Training Modes:

- Baseline Model Training: Initially, we evaluated the performance of the Orca2 model on each dataset section without any specific fine-tuning. This provided us with a baseline accuracy metric for comparison with subsequent experiments.

- Normal Fine Tuning: In this mode, we fine-tuned the Orca2 model on the respective datasets using conventional prompt tuning techniques. The model was trained to understand numeral-rich contexts and generated responses accordingly.

- Chain-of-Thoughts tuning Method: As an extension of traditional fine-tuning, we explored the Chain-of-Thoughts tuning method to train Orca2. This approach encourages the model to retain contextual information across sequential examples, enhancing its ability to grasp complex numeral-related nuances.

4 Experiments

4.1 Data

The four Used datasets (QP, QQA, QNLI [2] and AWPNLI [3]), are all related to natural language processing tasks and have been widely used in research and benchmarking for various NLP models, particularly those based on deep learning.

These datasets are often utilized to evaluate the performance of NLP models, particularly those designed for tasks like question answering, paraphrase detection and natural language inference. They provide standardized benchmarks for assessing the capabilities of different models and techniques in handling these tasks effectively.

4.2 Evaluation method

In evaluating the performance of the NumEval model across the QP, QQA and QNLI datasets, we employ a series of evaluation metrics tailored to the specific characteristics of each dataset and the different training modes applied to the Orca2 model.

4.2.1 Evaluation metrics:

Accuracy, F1-score and Recall serves as the primary evaluation metric across all experiments conducted on the QP, QQA, QNLI and AWPNLI datasets. It represents the proportion of correctly classified instances over the total number of instances in the datasets.

4.2.2 Experimental Modes:

Three experimental modes are considered in the evaluation:

- Basic Orca2 model without any fine-tuning.
- Orca2 model fine-tuned.
- Orca2 model fine-tuned using the Chain-of-Thought method.
- Sequential fine-tuning of Orca2 model using QP and QNLI datasets

Cross-dataset generalization is evaluated by training the Orca2 model on one dataset and subsequently fine-tuning it on another dataset to assess the model's ability to transfer knowledge across domains.

[2] https://drive.google.com/drive/folders/1mKbiL420U4Ih-hGmpaSki0FCvGHH3Au2?usp=sharing

[3] https://drive.google.com/file/d/10JNRN6iI5u9ZbEJPUEq4LAksBPW3vzGG/view?usp=sharing

By employing these evaluation methods, we aim to comprehensively assess the effectiveness of the NumEval framework in numeral-aware language understanding and generation tasks across diverse datasets and training modes.

4.3 Experimental details

In our study, we conducted experiments utilizing the NumEval framework, focusing on the QP, QQA and QNLI datasets to assess the performance of the Orca2 model. Below, we outline the experimental details for each dataset and the various modes of training and testing conducted.

4.4 Results

In this section, we present the quantitative results obtained from our experiments across the QP, QNLI and QQA datasets. We compare our results against baselines established by various models, including BERT, CN-BERT, LinkBERT, CN-LinkBERT, RoBERTa and CN-RoBERTa, each evaluated on different data modes: original, Digit-based and ScientificNotation.

In table 1 presents the powers claimed in the article.

In the following, we present the quantitative results of our experiments conducted on the QP, QQA and QNLI datasets using the Orca2 model in various training modes including normal Fine tuning and Chain-of-Thoughts tuning in table 2. Additionally, we discuss the implications of these results in relation to our initial hypotheses and the effectiveness of our approach.

In a series of experiments, the model was Fine Tuned and tested on QNLI dataset which has scientific numbers or numbers that the decimal part is removed by multiplying by a large number with multiples of 10 units.

The quantitative results of our experiments reveal several noteworthy findings. Firstly, in the QP dataset experiments, we observed a substantial improvement in accuracy from the baseline when employing both Normal Fine-tuning and Chain-of-Thoughts tuning methods. Notably, the Chain-of-Thoughts tuning approach yielded the highest accuracy at 97.25%, demonstrating the effectiveness of sequential reasoning in improving model performance.

In contrast, the experiments conducted on the QNLI dataset showed similar trends, with both normal Fine Tuning and Chain-of-Thoughts tuning methods outperforming the baseline accuracy. The Chain-of-Thoughts tuning method again exhibited superior performance, underscoring its efficacy in capturing nuanced relationships within the data.

We developed a classifier model capable of discerning prompts based on their respective dataset classes: QP, QQA, and QNLI. This classifier model serves to categorize prompts and subsequently directs them to the corresponding model tailored to handle the specific prompt class. This streamlined approach obviates the necessity for segregating the datasets, enhancing overall

Model	Mode	QP_Comment	QP_Headline	QNLI	QQA
BERT	Original	70.44%	57.46%	99.91%	53.20%
	Digit-based	65.38%	54.74%	99.11%	53.75%
	ScientificNotation	65.31%	55.99%	99.56%	53.24%
CN-BERT	Digit-based	69.93%	54.84%	99.42%	52.53%
	ScientificNotation	64.87%	56.40%	99.42%	66.63%
LinkBERT	Original	68.81%	55.70%	99.91%	54.14%
	Digit-based	63.76%	55.41%	99.73%	53.44%
	ScientificNotation	65.81%	56.05%	99.82%	54.33%
CN-LinkBERT	Digit-based	68.61%	54.44%	100%	50.44%
	ScientificNotation	63.48%	53.15%	99.73%	52.11%
RoBERTa	Original	60.46%	58.03%	98.93%	51.96%
	Digit-based	69.25%	57.65%	99.91%	51.96%
	ScientificNotation	64.32%	55.49%	100%	53.67%
CN-RoBERTa	Original	86.86%	77.29%	99.94%	50.71%
	Digit-based	64.25%	55.92%	99.73%	50.88%
	ScientificNotation	60.28%	54.85%	99.47%	52.27%

Table 1: Accuracy Results of article Models (Chen et al., 2023b)

Experiment	Training Mode	QP_Comment	QP_Headline
Experiment 1	Baseline	68.48%	80.12%
Experiment 2	Normal Fine Tuning	96.12%	97.65%
Experiment 3	Chain-of-Thoughts Tuning	75.83%	82.79%
Experiment 4	FT on QNLI of FT on QP	83.81%	82.58%

Table 2: Test results for the Orca2 model on the QP dataset

Experiment	Training Mode	Accuracy
Experiment 1	Baseline	31.82%
Experiment 2	Normal Fine Tuning 1 epoch	98.34%
Experiment 3	Normal Fine Tuning 2 epoch	99.52%
Experiment 4	Chain-of-Thoughts Tuning 1 epoch	58.19%
Experiment 5	Chain-of-Thoughts Tuning 2 epoch	61.32%
Experiment 6	Baseline Normal Fine Tuning on QP	32.82%
Experiment 7	Baseline Chain-of-Thoughts Tuning on QP	33.23%

Table 3: Test results for the Orca2 model on the QNLI dataset

Model	Accuracy (%)	F1 score (%)	Recall (%)
Normal Fine Tuning 1 epoch	98.34%	98.51%	98.34%
Normal Fine Tuning 2 epoch	99.52%	99.52%	99.53%
Chain-of-Thoughts Tuning 1 epoch	58.19%	55.1%	58.19%
Chain-of-Thoughts Tuning 2 epoch	61.32%	55.64%	61.32%

Table 4: F1 score and Recall for the Orca2 model on the Prompt Tuning by char-QNLI on QNLI dataset

Experiment	Training Mode	Accuracy (%)
Experiment 1	Normal Fine Tuning on statement-sci-10e 1 epoch	33.74%
Experiment 2	Normal Fine Tuning on statement-sci-10e 2 epoch	53.99%
Experiment 3	Normal Fine Tuning on char-QNLI 1 epoch	96.87%
Experiment 4	Normal Fine Tuning on char-QNLI 2 epoch	99.65%
Experiment 5	Normal Fine Tuning on char-QNLI 3 epoch	99.79%

Table 5: Test results for the Orca2 model on the Normal Fine Tuning by char-QNLI on QNLI dataset

Model	Accuracy (%)	F1 score (%)	Recall (%)
Normal Fine Tuning on char-QNLI 1 epoch	96.87%	96.86%	96.86%
Normal Fine Tuning on char-QNLI 2 epoch	99.65%	99.64%	99.64%
Normal Fine Tuning on char-QNLI 3 epoch	99.76%	99.79%	99.76%

Table 6: F1 score and Recall for the Orca2 model on the Fine Tuning by char-QNLI on QNLI dataset

Experiment	Training Mode	Accuracy (%)
Experiment	Baseline	100%

Table 7: Test Results for Orca2 in QQA Dataset

efficiency and coherence in the evaluation process.This structure is shown in Figure 1.

We employed a two-step fine-tuning process. Initially, the Orca2 model underwent fine-tuning on the QP dataset. Subsequently, we further fine-tuned the model using the QNLI dataset. The sequential fine-tuning approach allowed the model to adapt to the nuances of each dataset progressively.

We use the Orca2 model as an agent to assign each task to the specific agent. To achieve this task, we used a part of the available datasets to train our agent and achieved 99.4% accuracy in assigning tasks. As a result, we reached 96.13% accuracy on QP dataset, 100% accuracy on QQA and 98.85% accuracy on QNLI.

5 Analysis

Our analysis has illuminated the intricate nature of numeral-aware language tasks, emphasizing the imperative for ongoing scrutiny and enhancement of model architectures and training methods. The insights derived from our investigation notably elucidate the performance and adaptability of the Orca2 model within the NumEval task domain.

Upon reflection, we determined that optimizing the generality of the structure entails employing expert agents for individual sub-tasks, facilitated by a supervisory agent to assign tasks effectively. This strategic adjustment yielded heightened accuracy across all sub-tasks, marking a significant advancement in our approach.

6 Conclusion

In conclusion, our analysis highlights the intricate interplay between dataset characteristics, model architectures, and training methodologies in the NumEval task. While achieving significant advancements in numeral-aware language understanding and generation, our study underscores the importance of comprehensive evaluations encompassing both quantitative metrics and qualitative assessments to unravel the complexities of numerical reasoning in natural language understanding tasks. Moving forward, further research into adaptive learning strategies and nuanced dataset annotations promises to enrich our understanding and advancement in numeral-aware language processing tasks.

References

Semeval-2024 task 7: Numeral-aware language understanding and generation.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Nquad: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2925–2929.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023a. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

Chung-Chi Chen et al. 2023b. Improving numeracy by input reframing and quantitative pre-finetuning task. *Findings of the Association for Computational Linguistics: EACL 2023*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

A Mitra, L Del Corro, S Mahajan, A Codas, C Simoes, S Agarwal, X Chen, and A Razdaibiedina. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. Do language embeddings capture scales? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4889–4896, Online. Association for Computational Linguistics.

A Example Appendix

A.0.1 More about data set:

In each dataset there is a column called statement-char and statement-sci-10e. statement-char has spaced between the numbers inside the text and statement-sci-10e has written the numbers inside the text in scientific form (Chen et al., 2023b).

A.0.2 Additional Experiments:

We explored the integration of human-reasoning into the AWPNLI dataset by generating human-reasoning for 190 samples using ChatGPT.

Contrary to expectations, the inclusion of human-reasoning did not yield the anticipated accuracy improvements, highlighting the complexity of the task and potential limitations of our approach.

DUTh at SemEval-2024 Task 6: Comparing Pre-trained Models on Sentence Similarity Evaluation for Detecting of Hallucinations and Related Observable Overgeneration Mistakes

Ioanna Iordanidou **Ioannis Maslaris** **Avi Arampatzis**

Database & Information Retrieval research unit,
Department of Electrical & Computer Engineering,
Democritus University of Thrace, Greece.
{ioaniord1,imaslari,avi}@ee.duth.gr

Abstract

In this paper, we present our approach to SemEval-2024 Task 6: SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes, which aims to determine weather AI generated text is semantically correct or incorrect. This work is a comparative study of Large Language Models (LLMs) in the context of the task, shedding light on their effectiveness and nuances. We present a system that leverages pre-trained LLMs, such as LaBSE, T5, and DistilUSE, for binary classification of given sentences into 'Hallucination' or 'Not Hallucination' classes by evaluating the model's output against the reference correct text. Moreover, beyond utilizing labeled datasets, our methodology integrates synthetic label creation in unlabeled datasets, followed by the prediction of test labels.

1 Introduction

Hallucinations in machine generated text are cases when the model generates output that is partially or fully unrelated to the source sentence. While being a non-frequent phenomenon, it can dramatically impact the user experience and the trust toward the system. Hallucination rates in multiple models vary from 2.8 to 16.2 percent, according to the hallucination leaderboard created by Vectara hosted on HF and GitHub. [1] While the problem of hallucinations is known, it remains challenging, and one aspect of that is the absence of proper datasets. As a result, previous studies relied on scenarios where models are encouraged to hallucinate (Lee et al., 2018; Raunak et al., 2021; Müller et al., 2019; Voita et al., 2020; Zhou et al., 2020). However, it is uncertain if these approaches are effective in more natural, undisturbed environments (Guerreiro et al., 2022).

Recent research conducted in relatively clean settings (Guerreiro et al., 2022) demonstrates that existing hallucinations detection methods fall short. The authors create a natural setting dataset, annotate it for various NMT (Neural Machine Translation) pathologies, and evaluate detection methods. They find most existing detection methods inadequate, with sequence log-probability performing best. Although they demonstrated interesting results, they were limited on detecting hallucinations on Machine Translation (MT) generated text.

SemEval-2024 task 6 (Mickus et al., 2024) goes a step further by providing a human annotated dataset of hallucinated text regarding three different scenarios. Along with Machine Translation (MT), also Definition Modeling (DM) and Paraphrase Generation (PG) cases are considered. This paper describes the system developed by the DUTh team for SemEval-2024 task 6. Our strategy is based on utilizing embeddings to evaluate the similarity between context and hypothesis sentences in order to detect hallucinated text. In our case context sentence is the 'gold' output expected from to the models for generation and hypothesis sentence is the actual model production. For generating the embeddings of the context and hypothesis we are utilizing a pretrained T5 tokenizer (Raffel et al., 2020). Then we measure their similarity by taking the dot product of their corresponding embeddings, followed by summation along axis 1. Finally, using that similarity score, we train an ensembler machine learning model to distinguish hallucinated form non-hallucinated text. We provide our code publicly [2]

2 Background

2.1 Related work

Methods for identifying hallucinations are primarily concentrated on the Machine Translation task and they generally aim to find translations of poor quality that may also satisfy additional constraints. To effectively pinpoint factual inaccuracies in LLM

[1] https://huggingface.co/spaces/vectara/leaderboard

[2] https://github.com/DataMas/ai-hallutinations-detection

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1064–1070

outputs, one straightforward strategy involves comparing the model generated output information from an external knowledge source. Relevant research, starting from traditional fact checking (Augenstein et al., 2019) tries to expand the capabilities of such systems by incorporating various web sources (Chen et al., 2023) and evaluating their truthfulness (Galitsky, 2023). Recently, there is a significant emphasis on enhancing the process of retrieving information from external sources. FACTSCORE introduced by (Min et al., 2023), is a metric specifically for long-text generation. The LLM output is decomposed into atomic facts and each one is validated by reliable external knowledge sources. Furthermore, (Huo et al., 2023), enhanced the retrieving process by augmenting the query to the external sources with the input to and the output of the LLM.

When utilizing external sources, previous research mostly focused on evaluating models output based on a pool of third party knowledge. However, implementation of such systems could be complicated. Similarity between the source and the target estimated via embeddings, has been proved to be a good indicator for hallucinations in Machine Translation scenarios (Dale et al., 2022). In this manner, we are experimenting with this strategy on detecting hallucinations on machine generated text regarding Definition Modeling and Paraphrase Generation. We hypothesize that hallucinations can have a great impact on the conceptual content of the generated text, enough to be detected through sentence similarity evaluation.

2.2 Dataset

The task provided three datasets for each track: the train and test sets comprised unlabeled datapoints, and the validation set contained labeled datapoints enriched with additional features. Each datapoint in the labeled set encompasses the following attributes. The 'model' attribute is included only in the model-aware datasets and the items without bold annotation are not featured in the test datasets.

- **"id":** The datapoint's ID
- **"task":** The model's optimization objective (DM, PG, MT).
- **"model":** The model used for text generation.
- **"tgt":** The intended reference text for model generation.
- **"src":** The input presented to the models for generation.
- **"hyp":** The actual model output.
- **"ref":** Indicates whether the 'tgt' or 'src' fields, or both, are the context that contains the requisite semantic information to discern the datapoint as a hallucination.
- "labels": A set of per-annotator labels gauging whether each annotator perceives the datapoint as a hallucination.
- "label": The majority-based gold-label derived from the per-annotator labels.
- "p(Hallucination)": The probability assigned to the datapoint being a hallucination based on the proportion of annotators considering it as such.

In the model-aware segment, we dived deeper into our data by visually representing (Figure 1) the distribution of three distinct models across data points. In both the validation and test sets, two of the three models exhibit an equal distribution, while the third one, tuner007/pegasus_paraphrase, is utilized less, accounting for roughly 33 percent. The training set shows an equal distribution of all three models.

3 System Overview

3.1 Hallucination Detection

Hallucination detection methods are a developing field in modern NLP. Given input information and parameters can vary and subsequently the methods applied for detection are subject to change. SemEval-2024 Task 6: SHROOM - a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes was divided into two tracks. The first sub-task, Model-Aware, involves determining whether the model produced a hallucination, given information about which model was employed. The second sub-task, Model-Agnostic, pertains to scenarios where the model used is unknown.

3.2 System

We approached the task as a binary text classification problem, implementing our system leveraging the HuggingFace Transformer library. Concisely, our methodology aligns with the conventional approach to addressing text classification problems — training a model with a large labeled dataset and employing it to predict labels for the test set. We first opted for the labeled dataset provided by (Guerreiro et al., 2022) for the training process. Then

we tried the unlabeled training dataset supplied by the task organizers, conducting experiments to automatically generate synthetic labels for its utilization into the training process.

In summary, our system comprised distinct steps, including extraction of embeddings for 'hyp' and 'context', calculation of cosine similarity between the two, generation of synthetic labels for the training set through clustering, and prediction of the test set using ensembled classifiers.

3.2.1 Sentence Embeddings

Sentence embeddings are a potent technique utilizing deep learning models, specifically transformers, to encode words, or in our context, sentences, into vectors. These vectors capture the semantic meaning and contextual information of the input text. This encoding is valuable as vectors provide a robust representation of the semantic content embedded in sentences and are more efficiently compared or handled in any way for various NLP tasks. Our approach employed pre-trained sentence transformers sourced from the HuggingFace library (v. 2.2.2) for extracting these embeddings. From our dataset, the hypothesis ('hyp') was compared to the context sentence provided by the semantic reference ('ref'). The cosine similarity between the vectors resulting from this comparison, along with a probability measure of hallucination, was subsequently incorporated into our system. Formally, the similarity score, denoted as sims, is calculated as

$$\text{sims} = \sum_{i=1}^{n} (\text{emb\_con}_i \cdot \text{emb\_hyp}_i)$$

where emb_con_i and emb_hyp_i represent the embeddings for the i-th context and hypothesis, respectively. In this numerical measure of similarity, higher values indicate greater similarity between the encoded representations of hypotheses and contexts. The probability is computed as $1 - \text{sims}$ and is subsequently appended to the dataset.

3.2.2 Synthetic Labels Creation

Cluster analysis is a technique used in data mining and machine learning to group similar objects into clusters. k-means clustering is a popular unsupervised machine learning algorithm with vector inputs that allocates every data point to the nearest cluster. Synthetic data creation has become a widely adopted methodology within the NLP field, notably used for the purpose of label generation (Zhou et al., 2020). In our pursuit of generating synthetic labels for the unlabeled dataset, we employed the k-means algorithm with $k = 2$, signifying two centroids, to extract 'Hallucination' and 'Not Hallucination' labels. The parameters provided for clustering were the cosine similarity and the probability derived from sentence embeddings within the model-agnostic sub-task. Additionally, for the model-aware subtask, we incorporated the one-hot encoded representation of the utilized model as an additional parameter. We additionally tested the efficacy of our label extraction mechanism on the provided labeled datasets, achieving a notable accuracy rate of 75 percent.

3.2.3 Label Prediction

Following the training phase, we engaged in an ensemble approach, combining several widely recognized classification algorithms to forecast the labels of the test set and identify instances of hallucination. In text classification, each data point is allocated a label, with binary classification typically involving two labels (e.g., 0 and 1). Model ensembling aims to utilize the collective strength of various classifiers to maximize overall performance. We employed the similarity extracted from the embeddings and integrated it as a feature alongside the probability in the training of our classifiers. In our ensembling, we incorporated the following classifiers: Logistic Regression, Random Forest, Gradient Boosting, K Nearest Neighbours, XGBoost and Decision Tree. By employing this methodology, we got labeled test sets as the final outputs.

3.3 Models

Central to our system are pre-trained models from the sentence transformers library of HuggingFace (v. 2.2.2). The models we distinguished were DistilUSE (Reimers and Gurevych, 2019), LaBSE (Feng et al., 2020) and T5 (Raffel et al., 2020). The 'distiluse-base-multilingual-cased-v2' model excels at mapping sentences to a 512-dimensional dense vector space, making it ideal for tasks like clustering and semantic search. With its multilingual capabilities and nuanced representation of case information, it proves valuable across various languages for applications requiring semantic understanding and similarity assessment. LaBSE, Language-agnostic BERT sentence embedding, supports 109 languages and adopts a dual-encoder approach based on pretrained transformers. It has been fine-tuned for translation ranking with an ad-

ditive margin softmax loss. T5, or Text-To-Text Transfer Transformer, is adept at mapping sentences to a 768-dimensional dense vector space. This model particularly excels in tasks related to sentence similarity. This selection of models, ranging from BERT to LaBSE and T5, offers a diverse toolkit for our system. These pre-trained models, with their distinct architectures and capabilities, contribute to the robustness and versatility of the implemented system across a spectrum of natural language processing tasks.

Model	Model-Agnostic		Model-Aware	
	Score 1	Score 2	Score 1	Score 2
LaBSE	0.7366	0.7366	0.7440	0.7440
T5	0.7440	0.7440	0.7553	0.7553
DistilUSE	0.7066	0.7367	0.6867	0.7440

Table 1: Accuracy for all models. Score 1 is using synthetic labeled train set and score 2 is using the Guerreiro set.

Model	Model-Agnostic		Model-Aware	
	Score 1	Score 2	Score 1	Score 2
LaBSE	0.4298	0.4298	0.4277	0.4277
T5	0.4748	0.5224	0.5285	0.5255
DistilUSE	0.3051	0.3576	0.2988	0.3269

Table 2: Spearman Correlation for all models. Score 1 is using synthetic labeled train set and score 2 is using the Guerreiro set.

4 Experimental setup

4.1 Preprocessing

Prior to any NLP problem solving , performing text preprocessing is necessary. The nature of text preprocessing varies depending on the methodology to be employed, encompassing various steps. In the context of our binary classification problem utilizing the provided dataset, we conducted thorough feature extraction and preprocessing on the raw textual data. Specifically, we opted for the English language model available in SpaCy's trained pipelines (v. 3.7.2). This choice was particularly informed by the necessity to preprocess the 'hyp' and 'context' features, being aware that the context in the Machine Translation (MT) task was in English. Across all tasks, our text preprocessing included text lowercase conversion, punctuation removal, and lemmatization, where custom lemmas were incorporated. For the Definition Modeling task, we extracted the word to define from the context. In the model-aware track of the task, we introduced one-hot encoding representation of the model used for all datapoints. Similar techniques were used for the (Guerreiro et al., 2022) dataset adapting to the corresponding feature names.

4.2 Experiments

The conducted experiments incorporated the entirety of available datasets, the training, development, and test sets. Our initial experiment, as outlined in the system section, involved the utilization of the synthetically labeled train and test sets in conjunction with the DistilUSE, LaBSE, and T5 models. In the next experiment, we only utilized T5 and skipped the synthetic labeling phase from our methodology. In this iteration, the training process was conducted by utilizing the (Guerreiro et al., 2022) dataset. This adjustment was motivated by the labeled nature of this set, making it conducive to predicting labels for the test set in both tracks of the task.

4.3 Evaluation

The evaluation measures employed in both tracks of the task were consistent. The initial metric pertained to a general accuracy score, derived from the test reference data provided by the task organizers, applied to our binary classification results. Subsequently, the evaluation for the model-agnostic track extended to include the Spearman's correlation coefficient, a statistical measure of the strength of a monotonic relationship between the output probabilities of the systems and the proportion of annotators marking an item as overgenerating. The Spearman correlation assesses the degree to which the systems' output probabilities align with the consensus among annotators, offering a nuanced evaluation of the models' performance in capturing the observed trends in overgeneration perception. Both metrics have a maximum value of 1.

5 Results

The comprehensive scores of our system across the three utilized models are presented in Tables 1 and 2, for accuracy and correlation, respectively. The highest accuracy score was T5's 0.7553 in model-aware which ranked 25th out of 38 and 0.7440 in model-agnostic which ranked 27th out of 41 while both passed the baseline scores in accuracy and correlation. There was no difference in the dataset used for the training process. The baseline score was obtained through using an instruction-finetuned Mistral model tasked with classifying the sentences as contextual or not, answering with

Algorithm	Model-Agnostic		Model-Aware	
	Score 1	Score 2	Score 1	Score 2
Logistic Regression	0.6874	0.7066	0.7086	0.7160
Random Forest	0.6873	0.7420	0.7380	0.7347
Gradient Boosting Classifier	0.6874	0.7327	0.7067	0.7393
K Nearest Neighbours	0.6867	0.6740	0.7067	0.6687
Decision Tree	0.7067	0.7447	0.7367	0.7493
XGBoost	0.7067	0.6787	0.7407	0.6887
Ensembling	**0.7440**	0.7447	**0.7553**	0.7373

Table 3: Evaluation Metrics for Seven Machine Learning Algorithms. Score 1 is using synthetic labeled train set and score 2 is using the Guerreiro set.

a yes or no. The accuracy score it achieved was 0.697 in the model-agnostic track and 0.745 in the model-aware track. If ranked by correlation, T5 scores highest with a moderate correlation of 0.5285 for the aware track using the synthetically labeled dataset and 0.5224 in the agnostic track using the Guerreiro dataset, also surpassing the baseline system which scored 0.488 and 0.403 respectively. Consequently, after careful consideration, T5 was selected for integration into our final system. For the label prediction part we tried multiple Machine Learning classification algorithms which are shown in Table 3. In both tracks using the synthetic labeled dataset, distinctions, ranging from subtle in some cases to more pronounced in others, were observed among individual algorithms. However, the ensemble strategy consistently surpassed their individual performances, scoring 0.7440 in model-agnostic and 0.7553 in model-aware. When applying the Guerreiro dataset, Decision Tree outperformed the ensemble strategy in the model-aware track, consistently staying below 0.7553. However, this did not hold in the model-agnostic track, where both Decision Tree and the ensemble of all seven algorithms achieved a score of 0.7447, closely mirroring Score 1. Based on the previously mentioned outcomes, the score obtained through the ensemble of classifiers using the synthetically labeled set was ultimately submitted to the tasks leaderboard.

6 Conclusion

Through these experiments, we found that the pretrained T5 model exhibits optimal performance in the detection of hallucinated text in the domain of artificial intelligence. Furthermore, we successfully employed an ensemble of multiple popular top-tier classifiers to augment the predictive capabilities of our system and investigated the implications of synthetically labeling unlabeled data, presenting it as a novel approach to hallucination detection.

The next step could involve an extended comparison of various language models to identify the most powerful one, as well as exploring the option of training on diverse and larger datasets. Additionally, for further exploration, we recommend fine-tuning a Language Model (LLM) to extract enhanced embeddings, thereby improving the accuracy of sentence similarity assessments and consequently bolstering the overall system performance.

References

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. *arXiv preprint arXiv:1909.03242*.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2023. Complex claim verification with evidence retrieved in the wild. *arXiv preprint arXiv:2305.11859*.

David Dale, Elena Voita, Loïc Barrault, and Marta R Costa-jussà. 2022. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better. *arXiv preprint arXiv:2212.08597*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Boris A Galitsky. 2023. Truth-o-meter: Collaborating with llm in fighting its hallucinations.

Nuno M Guerreiro, Elena Voita, and André FT Martins. 2022. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. *arXiv preprint arXiv:2208.05309*.

Siqing Huo, Negar Arabzadeh, and Charles LA Clarke. 2023. Retrieving supporting evidence for llms generated answers. *arXiv preprint arXiv:2306.13781*.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Mathias Müller, Annette Rios, and Rico Sennrich. 2019. Domain robustness in neural machine translation. *arXiv preprint arXiv:1911.03109*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. *arXiv preprint arXiv:2104.06683*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2020. Analyzing the source and target contributions to predictions in neural machine translation. *arXiv preprint arXiv:2010.10907*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.

A Appendix

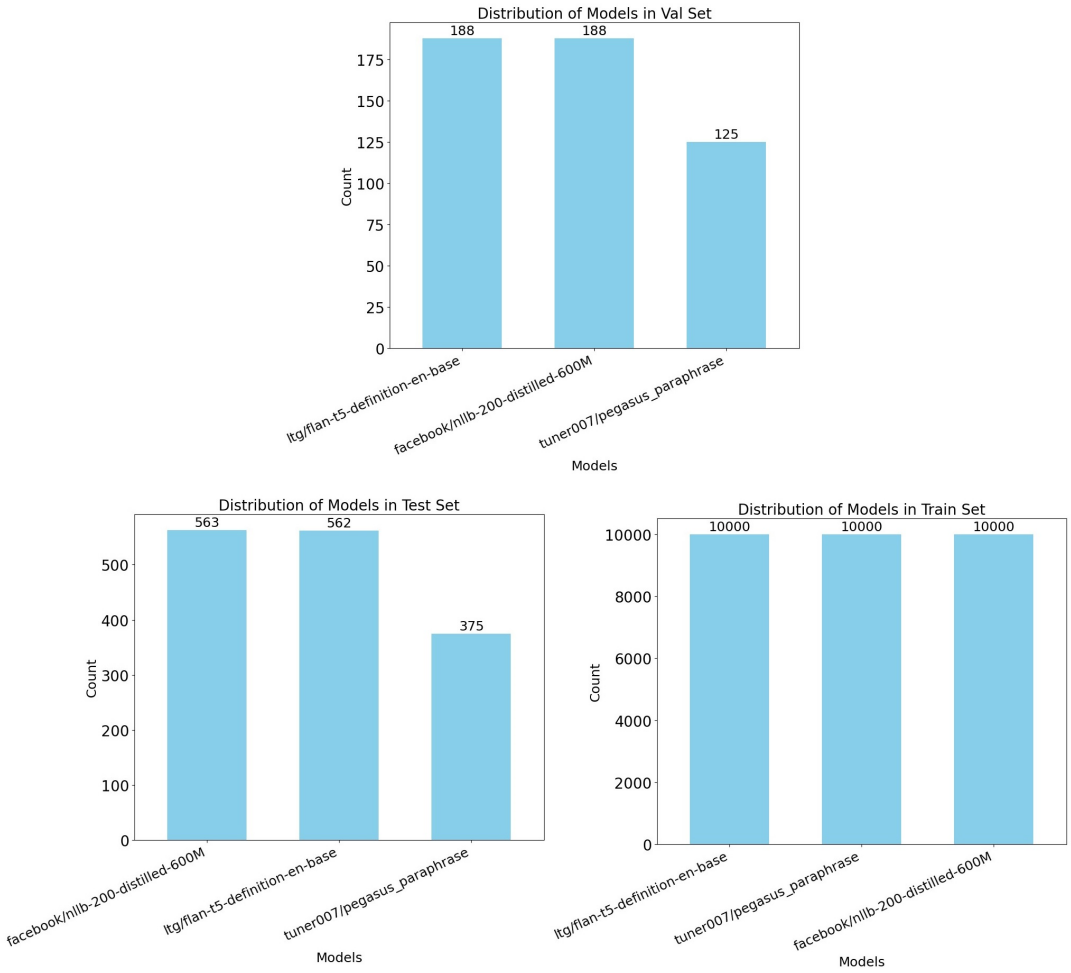


Figure 1: Distribution of Models used in Datasets.

MBZUAI-UNAM at SemEval-2024 Task 1: Sentence-CROBI, a Simple Cross-Bi-Encoder-Based Neural Network Architecture for Semantic Textual Relatedness

Jesus-German Ortiz-Barajas
Mohamed Bin Zayed University of Artificial Intelligence
Jesus.OrtizBarajas@mbzuai.ac.ae

Gemma Bel-Enguix
Instituto de Ingeniería
Universidad Nacional Autónoma de México
gbele@iingen.unam.mx

Helena Gómez-Adorno
IIMAS
Universidad Nacional Autónoma de México
helena.gomez@iimas.unam.mx

Abstract

The Semantic Textual Relatedness (STR) shared task aims at detecting the degree of semantic relatedness between pairs of sentences on low-resource languages from Afroasiatic, Indoeuropean, Austronesian, Dravidian, and Nigercongo families. We use the Sentence-CROBI architecture to tackle this problem. The model is adapted from its original purpose of paraphrase detection to explore its capacities in a related task with limited resources and in multilingual and monolingual settings. Our approach combines the vector representation of cross-encoders and bi-encoders and possesses high adaptable capacity by combining several pre-trained models. Our system obtained good results on the low-resource languages of the dataset using a multilingual fine-tuning approach.

1 Introduction

Task 1 of SemEval 2024 (Ousidhoum et al., 2024b) focuses on Semantic Textual Relatedness (STR). Given two sentences, the semantic relatedness between them is defined as the degree of closeness between their meanings (Mohammad and Hirst, 2012). However, the traits that make two sentences to be understood as related entities can be of different order, such as the underlying syntactic structure, lexical affinity, or the author's style, among others.

The task organisers have chosen for this track a set of languages, among which English and Spanish stand out, two languages with numerous computational resources. The rest, are low-resourced languages from Africa (Algerian Arabic, Moroccan Arabic, Amharic, Hausa, Kinyarwanda) and Asia (Marathi, Telegu).

Three tracks were proposed in the task: supervised, unsupervised and cross-lingual. We participated in Track A, supervised. This is a regression problem since a relatedness coefficient must be given that ranges from 0 to 1 for each pair of sentences. Our solution is based on using the sentence-CROBI model, introduced in Ortiz-Barajas et al. (2022), which was designed for paraphrase detection with very good results in English. Our hypothesis is that the same methods used in paraphrase detection can be applied to the determination of the degree of relatedness.

The structure of the paper is the following. In section 2, we describe the related work on this task using pre-trained language models. Section 3 briefly describes the dataset. In section 4, we present our methodology. Finally, we present our results in the development and evaluation phases in section 5 and conclusions in section 6.

2 Related Work

The Sentence-BERT model (Reimers and Gurevych, 2019) is an approach that generates semantically meaningful sentence embeddings. By training BERT on siamese and triplet network structures, this approach is able to capture sentence similarity more effectively. It also reduces computational overhead compared to BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) while maintaining high accuracy in tasks such as semantic textual similarity and transfer learning.

Following this research line, there is an approach to improve BERT-based semantic embeddings for similarity tasks (Li et al., 2020). The authors propose a flow-based calibration method by transforming the original BERT embeddings into an isotropic latent space using flow. The proposed method aligns better with gold semantic similarity and reduces the influence of lexical similarity.

In this work, we use the Sentence-CROBI model, a simple architecture that combines bi-encoders and cross-encoders that was originally proposed to solve paraphrase detection. Due to its implementation facility, we adapt this model for the semantic relatedness task by only changing the task-specify

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1071–1079

Language	train	dev	test
Amharic (amh)	992	95	171
Algerian Arabic (arq)	1,262	92	584
Moroccan Arabic (ary)	925	70	427
English (eng)	5,500	250	2,500
Spanish (esp)	1,562	140	600
Hausa (hau)	1,763	212	603
Marathi (mar)	1,155	293	298
Telugu (tel)	1,146	130	297
Kinyarwanda (kin)	778	102	222

Table 1: Number of instances in each train, dev and test language partition for the supervised learning track of the SemRel dataset.

block, the loss function and the pre-trained models for the cross-encoder and bi-encoder components.

3 Corpora

We briefly describe the corpora that we use to evaluate our model in the SemEval shared task 1 in this section.

The SemRel2024 dataset (Ousidhoum et al., 2024a) is a comprehensive collection of semantic textual relatedness datasets for 14 languages, predominantly spoken in Africa and Asia. These languages cover a wide range of language families and include both high-resource and low-resource languages. Each dataset consists of sentence pairs annotated by native speakers with relatedness scores ranging from 0 (completely unrelated) to 1 (maximally related). The datasets were curated by selecting pairs from various sources such as news data, Wikipedia, and conversational data to ensure diversity in topics and formality levels. The relatedness scores were generated through Best-Worst Scaling (BWS) annotations, enhancing the reliability of the rankings. Table 1 shows the SemRel dataset statistics for all languages in the supervised learning track.

It can be noticed it is a highly unbalanced dataset. Only English has more than 2,000 training examples, followed by Hausa, Spanish, Algerian Arabic, Marathi and Telugu with more than 1,000 instances and Amharic, Moroccan Arabic and Kinyarwanda with less than 1,000 examples.

4 Methodology

In this section, we describe the proposed architecture, the experimental configuration and the training details. For pre-processing the sentence pairs, we perform the same text pre-processing steps as mentioned in (Ortiz-Barajas et al., 2022).

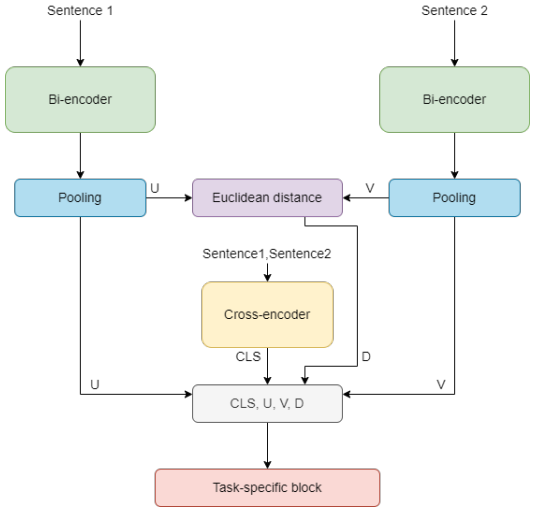


Figure 1: Diagram of the Sentence-CROBI model. U and V correspond to the individual vector representation of each text, CLS is the token classification obtained with the cross-encoder, and D is the Euclidean distance between U and V

4.1 Model

In this section, we present the Sentence-CROBI (Ortiz-Barajas et al., 2022) architecture and its implementation. The model has two main components: a bi-encoder and a cross-encoder. The bi-encoder is based on the Sentence-BERT model (Reimers and Gurevych, 2019); this is a BERT modification using a Siamese neural network that enables the model to obtain single vector representations for each text by applying a Pooling operation to the last hidden state of the bi-encoder model. We represent these vectors as u and v, respectively. The cross-encoder component receives the joint encoding of the sentence pair and is capable of capturing the relation between both texts. We use the classification token [CLS] as a final vector representation of the sequence.

We obtain a global representation of the sentence pair by concatenating the classification token [CLS] from the cross-encoder representation, the Euclidean distance D between u and v vectors, and the vectors u and v itself. This global vector is the input to a task-specific block composed of two fully connected networks with a single-neuron output. Figure 1 shows the structure of the Sentence-CROBI model.

The output of the bi-encoder component is a contextualised word embedding matrix obtained

by taking the last hidden state of the component, where each row represents a word of the input sentence. In this work, we apply a mean Pooling operation, averaging all the matrix dimensions to obtain a vector representation.

Since we are working on a regression problem, the task-specific layer of our model is composed of a fully connected network featuring two layers. Initially, it accepts the global representation of sentence pairs as input, undergoing a Dropout (Hinton et al., 2012) layer with a probability of 0.1. This regularisation technique is implemented to prevent network over-fitting by randomly zeroing some input values. Subsequently, the input proceeds through a fully connected layer of 1793 units, employing a hyperbolic tangent as the activation function. Ultimately, the output layer is composed of one neuron.

We use the Mean Squared Error (MSE) as a loss function during the training of the Sentence-CROBI model. MSE quantifies the average squared difference between the predicted values and the ground truth across a dataset, which is widely used in deep learning (Bishop, 2006; Goodfellow et al., 2016). For a dataset with N samples, MSE is defined as the mean of the squared differences between predicted $\hat{y}_i$ and actual y_i values as shown in 1.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (1)$$

Notably, our task-specific block and the loss function differ from that proposed in (Ortiz-Barajas et al., 2022) as paraphrase detection entails a binary classification task. In contrast, semantic relatedness is defined as a regression task.

One of the advantages of the Sentence-CROBI model is its implementation facility that only relies on using two pre-trained models, one as a bi-encoder and the other as a cross-encoder. The selection of these models depends on the specific task and available computational resources. The implementation facility also allows the performing of fast experimentation with minor changes. These model features enable us to build solutions for all languages in Track A following the same methodology.

4.2 Data splitting

We perform K-fold cross-validation to create training and validation subsets in the development phase of the shared task. The process entails iteratively designating one of the K folds as the validation set while the remaining $K-1$ folds collectively form the training set. This procedure is repeated K times, with each of the K folds serving as the validation set exactly once. K-fold cross-validation mitigates the impact of data partitioning on model assessment and aids in obtaining a more reliable estimate of a model's performance (James et al., 2013). We set $K = 5$ for all languages in the dataset.

4.3 Fine-Tuning

In this section, we describe our fine-tuning approaches. All approaches use a small number of epochs and a small learning rate. We train our models with a batch size of 32, a learning rate in the range $\{1 \times 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}\}$, and the Adam optimizer (Kingma and Ba, 2014), with a warm-up ratio of 0.06 and a linear decay to zero. We train all models for a maximum of 10 epochs and perform pseudo early stopping to use the model with the best performance on the validation data. The maximum length is 35 for individual texts and 128 for text pairs. The tokenization method differs between sentence pairs and individual texts, resulting in varying length representations. Hence, the length of each representation does not align. We use HuggingFace's Transformers library (Wolf et al., 2020) to implement the Sentence-CROBI model. Our implementation is publicly available on GitHub[1].

The first experimental setting that we use follows a monolingual approach, which means we fine-tune a model for each language of the dataset. We leveraged the HuggingFace Hub platform[2] to select bi-encoder and cross-encoder components for each model. To constrain the search space, we exclusively focused on encoder-only architectures that were either pre-trained or fine-tuned for the specific language of interest and possessed an associated paper describing the employed dataset and training details. In case there are no specific-language models, we use a multilingual model. We provide further details for the bi-encoder and cross-encoder combinations for each language in the dataset to fine-tune our model in Appendix A.

We also follow a multilingual approach to fine-tune our model. We group the languages based on their linguistic family. We consider two families.

[1] https://github.com/jgermanob/Sentence-CROBI
[2] https://huggingface.co/models

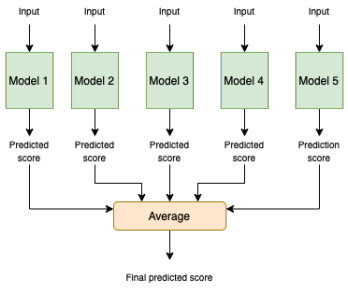


Figure 2: Bagging method diagram to obtain the final predicted score. Each model is fine-tuned using a different random seed and the final prediction is the average of all predictions.

The first one is the Semitic family, which includes Algerian and Moroccan Arabic as well as Amharic. The second one is the Indoeuropean family, which includes English, Spanish and Marathi. Telugu, Kinyarwanda and Hausa languages belong to different families; therefore, we do not include them in this approach. We concatenate each training and validation split to create each family-based split to train the models. For both families we use XLM-RoBERTa base (Conneau et al., 2020) as a cross-encoder and the multilingual uncased base version of BERT (Devlin et al., 2018) as a bi-encoder.

4.4 Ensemble Learning

In order to enhance the performance of the model in the Semantic Textual Relatedness task, we employ the Bagging method (Breiman, 1996), a strategy that mitigates generalisation errors by combining multiple models. This approach involves training different models independently and combining each output set to vote on test data and obtain the final prediction.

In the case of neural networks, differences in random initialisation or in batch generation cause independent errors in each member of the ensemble; therefore, the ensemble will perform significantly better than its members (Goodfellow et al., 2016).

We compute the final similarity score by averaging the output of each fine-tuned model with a different fold from the cross-validation splitting. Therefore, we use five distinct and independent models to obtain a final prediction. Figure 2 shows a diagram of how this method is used in this work.

5 Results

We present the results of our proposed model in the following section in the development and evaluation phases of the SemEval 2024 Task 1: Semantic Textual Relatedness.

Lang	Val ρ (avg)	Dev ρ
amh	0.4828	0.6230
arq	0.4784	0.6370
ary	0.7308	0.8030
eng	0.8709	0.8440
esp	0.5861	0.6900
hau	0.6076	0.6740
mar	0.7913	0.8470
tel	0.7290	0.8112

Table 2: Results of the proposed model in the development phase using a monolingual fine-tuning approach. We report an average of 5 runs in the validation splits used for cross-validation. We obtain the final score predictions in the development set using the bagging technique.

5.1 Development Phase

We report the average Spearman rank correlation coefficient in the validation dataset corresponding to each fold and the performance score in the development dataset reported in the Codalab page of the shared task for the development phase. We obtain the final score for each instance in the development dataset using the bagging technique and the average predictions of the five independent models for each fold.

In the case of the monolingual fine-tuning approach, we use a different model for each language in the dataset. Table 2 shows the results for each language. Half of our results in this approach achieve a performance higher than 0.80 in the performance metric, while the remaining models obtain a result above 0.60. The best performance is for the English language, with a Spearman correlation coefficient of 0.844. In contrast, the lowest performance is for the Amharic language, with a Spearman correlation coefficient of 0.623.

Due to the imbalance present in the dataset, we employed a multilingual fine-tuning approach by grouping languages into linguistic families. In this approach, we considered two groups: the Semitic (Sem) languages and the Indoeuropean (IE) languages.

Table 3 shows the results using the multilingual fine-tuning approach. There is a performance decrease in 6 of 8 considered languages. In the case of the Indoeuropean family, our model obtains a Spearman correlation coefficient of 0.8191 for En-

Lang	Fam	Val ρ (avg)	Dev ρ
eng	IE	0.8079	0.8191
esp	IE	0.8079	0.6874
mar	IE	0.8079	0.8290
amh	Sem	0.6926	0.8223
arq	Sem	0.6926	0.4727
ary	Sem	0.6926	0.8519

Table 3: Results of the proposed model in the development phase using a multilingual fine-tuning approach. We report an average of 5 runs in the validation splits used for cross-validation. We obtain the final score predictions in the development set using the bagging technique.

glish, which represents a 0.0249 decrease; in the case of Spanish and Marathi, our model decays 0.0026 and 0.018, respectively. In the case of the Semitic family, the multilingual fine-tuning approach improves the model performance in 2 of 3 considered languages: Moroccan Arabic (Moroc. A.) and Amharic. The model increases its performance from 0.803 to 0.8519 in Moroccan Arabic and from 0.623 to 0.8223 in Amharic, which represents a 0.1993 performance improvement in terms of Spearman correlation coefficient.

We must mention that we did not report any results for the Kinyarwanda language in the development phase because it was added to Track A later (December 12, 2024). Therefore, we were unable to conduct any experiments prior to the evaluation phase.

5.2 Evaluation Phase

We select the best-performing model for each language in the evaluation phase of the shared task. We use a monolingual fine-tuning approach for Algerian Arabic, English, Spanish, Hausa, Marathi, Telugu and Kinyarwanda, as well as a multilingual approach for Amharic and Moroccan Arabic. We create a new training set for each language and family by adding the development subset and its gold scores released by the shared task organisers. We train five independent models for each language and obtain the final score predictions using the bagging technique.

Table 4 shows the results of our proposed model in the evaluation test, its comparison with the baseline and the final ranking in the shared task for each language. We add a * to denote a multilingual-fine-tune-based approach. Our model outperforms the baseline in English and Moroccan Arabic with a difference from the leaders of 0.0183 and 0.0765, respectively. The lowest performance of the proposed model is in the Kinyarwanda language, with a Spearman correlation coefficient of 0.4585 and a difference from the leader of 0.3584.

Lang	Score	Baseline	Rank	Highest score
amh *	0.8398	0.85	7/18	0.8886
arq	0.5407	0.6	11/24	0.6823
ary *	0.7861	0.77	13/23	0.8625
eng	0.8316	0.83	16/36	0.8499
esp	0.6968	0.7	11/25	0.7403
hau	0.6702	0.69	9/21	0.7642
mar	0.8669	0.88	11/25	0.9108
tel	0.7847	0.82	17/25	0.8733
kin	0.4585	0.72	16/21	0.8169

Table 4: Results of the proposed model in the evaluation phase using monolingual and multilingual fine-tuning approaches compared with the baseline and the highest score. We obtain the final score predictions in the development set using the bagging technique. * Denotes a multilingual approach.

We perform an error analysis of our model's performance in the evaluation dataset for each language in Appendix B. The analysis suggests that the global vector representation of the sentence pair has a limited capacity to capture other semantic relationships between the texts apart from similarity, and future work should follow this direction. Nevertheless, it is essential to highlight that only the task-specific block should change, which illustrates the high adaptability capacity of the model.

6 Conclusions

This work presents the Sentence-CROBI model and its adaptation to the SemEval 2024 Task 1: Semantic Textual Relatedness. We evaluate the model's capacities in monolingual and multilingual fine-tuning approaches to measure its performance and adaptability across diverse linguistic families, yielding acceptable performance in low and mid-resource languages. Ensemble techniques further enhance the robustness and reliability of the model's predictions. Overall, the findings underscore the model's capacity for solving relatedness detection tasks, emphasising its versatility in accommodating linguistic variations and resource constraints.

References

Christopher Bishop. 2006. Pattern recognition and machine learning. *Springer google scholar*, 2:5–43.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. Deep learning, volume 1.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. 2013. *An introduction to statistical learning*, volume 112. Springer.

Raviraj Joshi. 2022. L3cube-mahacorpus and mahabert: Marathi monolingual corpus, marathi bert language models, and resources. In *Proceedings of The WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 97–101, Marseille, France. European Language Resources Association.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Javier De la Rosa, Eduardo G. Ponferrada, Manu Romero, Paulo Villegas, Pablo González de Prado Salas, and María Grandury. 2022. Bertin: Efficient pre-training of a spanish language model using perplexity sampling. *Procesamiento del Lenguaje Natural*, 68(0):13–23.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Saif M. Mohammad and Graeme Hirst. 2012. Distributional measures of semantic distance: A survey. *CoRR*, abs/1203.1858.

Jesus-German Ortiz-Barajas, Gemma Bel-Enguix, and Helena Gómez-Adorno. 2022. Sentence-CROBI: A simple cross-bi-encoder-based neural network architecture for paraphrase identification. *Mathematics*, 10(19):3578.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Hariom A. Pandya, Bhavik Ardeshna, and Dr. Brijesh S. Bhatt. 2021. Cascading adaptors to leverage english data to improve performance of question answering for low-resource languages.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen,

Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Seid Muhie Yimam, Abinew Ali Ayele, Gopalakrishnan Venkatesh, Ibrahim Gashaw, and Chris Biemann. 2021. Introducing various semantic models for amharic: Experimentation and evaluation with multiple tasks and datasets. *Future Internet*, 13(11).

A Monolingual approach

We use a monolingual fine-tuning approach for Track A, which means we fine-tune a model for each language in the dataset as described in section 4.3. We only consider publicily avaliable models in the HuggingFace Hub [3] that were either pre-trained or fine-tuned for the specific language and possess and associated paper describing the dataset as well as the training details.

Table 5 shows the bi-encoder and cross-encoder combinations for each language in the dataset to fine-tune our model following the monolingual approach. Following (Ortiz-Barajas et al., 2022) methodology, we choose a RoBERTa-based model for the cross-encoder and a BERT-based model for the bi-encoder. Only in the case of the Hausa language do we use a multilingual combination of bi-encoder and cross-encoder models because there are no available pre-trained or fine-tuned models that made our criteria.

B Error Analysis

We perform an error analysis of our model's performance in the evaluation dataset for each language. It is essential to mention that Spanish is excluded because the organisers do not provide the gold scores for this language.

Table 6 shows the differences between our model's predictions and the gold scores for each language in the evaluation dataset. We compute the difference by subtracting each example's predicted score from the gold score. Therefore, a negative difference means that our model predicts a higher score than the gold score, whereas a positive difference means that our model predicts a lower score than the gold score. The negative differences are higher than the positive differences in all languages. This result indicates that our model predicts a higher semantic textual relatedness score than the actual relatedness score in all cases.

Table 7 shows the top-5 negative differences predicted by the Sentence-CROBI model in the English evaluation dataset; that is, the model predicts a higher score than the gold score. It is possible to observe a high semantic similarity between the texts in the first four examples, and they can be considered paraphrases. Therefore, our model captures only one kind of semantic relatedness in these examples.

Table 8 shows the top-5 positive differences predicted by the Sentence-CROBI model in the English evaluation dataset; that is, the model predicts a lower score than the gold score. It is possible to observe different types of semantic relatedness that differ from semantic similarity between the texts. In the first example, the texts are semantic contrastive; the first text hints at excitement, while the second portrays boredom. The texts in the second example describe similar situations where a person performs some public activity. In the third example, both texts offer insights into events or situations concerning government or administration within a specific historical context. The semantic relatedness between the texts in the fourth example is their shared focus on the reading experience and the consideration of delving into further books within a series. Finally, the semantic relatedness in the fifth example lies in their depiction of situations involving young children, albeit with distinct tones and activities.

[3] https://huggingface.co/models

Lang	cross-encoder	bi-encoder
amh	Am-RoBERTa (Yimam et al., 2021)	mBERT-base FT on amharic-CC100 (Conneau et al., 2020)
arq	XLM-RoBERTa-base Arabic (Pandya et al., 2021)	BERT-base Arabic (Safaya et al., 2020)
ary	XLM-RoBERTa-base Arabic (Pandya et al., 2021)	BERT-base Arabic (Safaya et al., 2020)
eng	RoBERTa-large (Liu et al., 2019)	BERT-base (Devlin et al., 2018)
esp	BERTIN (la Rosa et al., 2022)	BETO (Cañete et al., 2020)
hau	XLM-RoBERTa-base (Conneau et al., 2020)	mBERT-base (Devlin et al., 2018)
mar	Marathi-RoBERTa (Joshi, 2022)	Marathi-BERT (Joshi, 2022)
tel	XLM-RoBERTa-base (Conneau et al., 2020)	Telugu-BERT (Joshi, 2022)

Table 5: Bi-encoder and cross-encoder model combinations for each language in the dataset using a monolingual fine-tuning approach.

Lang	Negative difference	Positive difference
amh	111	60
arq	335	246
ary	225	201
eng	1604	996
hau	314	289
kin	314	289
mar	238	60
tel	167	130

Table 6: Negative and positive differences in the scores predicted by our model concerning the gold score in the evaluation dataset for each language. A negative difference means that our model predicts a higher score than the gold score, whereas a positive difference means that our model predicts a lower score than the gold score.

Text 1	Text 2	Pred score	Gold score	abs diff
In general conversation , aerosol usually refers to an aerosol spray can or the output of such a can	When they say aerosol most people mean an aerosol spray can or the spray it makes	0.8610	0.44	0.4210
Ciampi was born in Livorno(Province of Livorno)	Carlo Azeglio Ciampi was born in 1920 in Livorno , Italy	0.7860	0.39	0.3960
TAKE A Shower then talk to her	I advise you to have a shower before speaking with her	0.8354	0.44	0.3954
if there 's a reason , we 'll discuss it	if you have a legitimate reason , we will discuss it	0.9060	0.52	0.3860
Forget that this is YA lit and READ IT	It's OK for what it is but you definitely won't forget you're reading a YA novel	0.7010	0.32	0.3810

Table 7: Top-5 negative differences predicted by the Sentence-CROBI model in the English evaluation dataset; that is, the model predicts a higher score than the gold score.

Text 1	Text 2	Pred score	Gold score	abs diff
A lot of this book is setting up the last book	This book is beige wallpaper	0.2798	0.64	0.3602
A man with glasses is playing his instrument in a small crown of people that includes another man in a suit with a trumpet	A man holding his arms out horizontally, and gripping a fencing sword in his right hand, as people in the background do the same thing	0.3780	0.69	0.3120
This date was January 3, 1867, which was two weeks before the beginning of the first administrative year of Governor Gove Saulsbury	Currently the distribution of the Senate Assembly seats was made to three senators for each of the three counties	0.2890	0.60	0.3110
i found it different from many other books i've read	I am trying to decide whether to read the other books in the series	0.4192	0.072	0.3008
A young boy wearing a red winter coat is eating and holding up a candy bar	A young baby boy crying while wearing a shirt that says ""I am the BOSS	0.3433	0.63	0.2867

Table 8: Top-5 positive differences predicted by the Sentence-CROBI model in the English evaluation dataset; that is, the model predicts a lower score than the gold score.

DUTh at SemEval 2024 Task 8: Comparing classic Machine Learning Algorithms and LLM based methods for Multigenerator, Multidomain and Multilingual Machine-Generated Text Detection

Theodora Kyriakou **Ioannis Maslaris** **Avi Arampatzis**
Database & Information Retrieval research unit,
Department of Electrical & Computer Engineering,
Democritus University of Thrace, Greece
{theokyri6,imaslari,avi}@ee.duth.gr

Abstract

Text-generative models evolve rapidly nowadays. Although, they are very useful tools for a lot of people, they have also raised concerns for different reasons. This paper presents our work for SemEval2024 Task-8 on 2 out of the 3 subtasks. This shared task aims at finding automatic models for making AI vs. human written text classification easier. Our team, after trying different preprocessing, several Machine Learning algorithms, and some LLMs, ended up with mBERT, XLM-RoBERTa, and BERT for the tasks we submitted. We present both positive and negative methods, so that future researchers are informed about what works and what doesn't.

1 Introduction

LLMs are becoming more and more part of our everyday lives due to their easy accessibility and their remarkably fluent responses in different fields like news, healthcare and education. This extensive usage can lead to unintended consequences. Specifically, LLMs could replace humans, provide sometimes false, incomplete or even misleading information, risk the critical thinking of students and progressively of the whole society. So, it is of high importance to find a way to identify if a text was written by a human or by a machine. Since all these complex models are trained on large datasets and have achieved generating texts that are so human like, it is difficult for a person to identify who generated a text. Here comes the importance of the automatic models, capable of differentiating between human written texts and machine generated texts, by exploiting patterns invisible to a human.

In this paper we describe the DUTh participation in *SemEval 2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection*. The task features three directions: *Binary Human-Written vs. Machine-Generated Text Classification, Multi-Way Machine-Generated Text Classification and Human-Machine Mixed Text Detection*. The first two refer to the scenario where a system must classify input texts which are fully written by a human or a machine. One detail regarding the second scenario is that we are also provided with the specific language model which generated the input text. In the third scenario we are presented with a text which is half human and half machine written and we have to determine the boundary, where the change occurs. In all subtasks the text data are coming from different sources and different generators, and we are not allowed to use any external data except for the ones given from the organizers.

The sub-tasks can be briefly described as follows: SubtaskA monolingual: given only English texts, we need to determine whether a text is human-written or machine-generated.

SubtaskA multilingual: given texts from 8 different languages (English, Arabic, Chinese, Indonesian, Urdu, German, Bulgarian, Russian), we need to determine whether a text is human-written or machine-generated.

SubtaskB: given only English texts, we need to determine whether a text is human-written or machine-generated and which is the specific generator.

SubtaskC: given only English mixed texts, where the first part is human-written and the second part is machine-generated, we need to determine the boundary.

Our team participates by submitting on SubtaskA (both monolingual and multilingual) and SubtaskB. During the competition we examine several methods, especially on SubtaskA monolingual, like different preprocessing techniques on the text data, several Machine Learning Algorithms, some ensembling methods and LLMs. We ended up submitting LLMs to all subtasks.

The models we choose are mBERT for subtaskA monolingual, XLM-RoBERTa for subtaskA multi-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1080–1086

lingual and BERT for subtaskB, as they achieve better performance on average. All these pre-trained models have been proven to be powerful for different NLP tasks.

Our proposed system for every subtask is a classifier based on a fine-tuned Large Language Model. During the training process, our model is provided with a text as input and a label regarding whether this input is human or machine generated. Additionally, this paper provides a comparative study of different LLMs fine tuned in this task. In this context, we also provide results regarding the use of classic machine learning algorithms trained to tackle this task.

2 Background

2.1 Dataset

All datasets given from organizers are on jsonl format. For both subtaskA and subtaskB the English human-written texts are coming from the following five sources, "wikihow", "wikipedia", "peerread", "reddit" and "arxiv". The generators for machine-generated texts are "chatGPT", "cohere", "davinci", "bloomz" and "dolly".

For multilingual data, the sources and generators are the same. The languages it consists of are English, Arabic, Bulgarian, Chinese, Indonesian, Urdu, German and Russian.

More information about the datasets and tasks can be found from the organizers.(Wang et al., 2024a) (Wang et al., 2024b) (Wang et al., 2024c)

2.2 Evaluation Metrics

The evaluation metrics for this task are accuracy, micro-f1 and macro-f1. Though, the organizers ranked both on validation and test set the participants basically based on the accuracy scores.

3 System Overview

Transformers have achieved state-of-the-art(Wolf et al., 2020) performances on several natural language processing tasks such as text classification. This is why all final models submitted are LLMs. Here we present the submitted model on each subtask.

We have all the hyperparameters for tuning the models submitted in the appendix section 6.

3.1 Tokenization applied

In all three models we use the tokenizer they already have. We define a max length of 512 tokens, which means that each encoder will take the first 512 tokens of the text as an input. We use truncation and padding, so that if a text has more than 512 tokens it gets cut off on the 512th token and if it has less than 512 tokens it gets padded until it reaches 512. We want all texts to have the same length. All the encoders of the transformers can give a representation of their input tokens, in a high dimensional space (512D here), based on the meaning of each token. For example, the same word can have different representation if its meaning changes. There is no other preprocessing made on the texts except for the tokenization applied by each model.

3.2 SubtaskA models

For the monolingual part of this subtask, we select multilingualBERT (bert-base-multilingual-cased)(Devlin et al., 2018). After comparing lots of classifiers, the two most performing are BERT and multilingualBERT. Previous research finds that there is no apparent benefit in training dedicated monolingual models for single language tasks, and actually by using a multilingual model instead may yield slightly improved performance de Vargas Feijó and Moreira (2007). Our case is no different. We can see that multilingualBERT slightly outperforms BERT on Table 8. MultilingualBERT is a pretrained model on 104 languages and has 179M parameters. For the multilingual part, we select XLM-RoBERTa (xlm-roberta-base)(Conneau et al., 2019) as it demonstrates the best performance between all models we examine. We do not apply any preprocessing on the input text, so XLM-R gets used as a cased model. XLM-R is pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages and has 279M parameters.

3.3 SubtaskB model

Between ML algorithms and LLMs we select BERT (bert-base-cased) for this task as we have seen that transformers, most of the times, have better results. It is pre-trained on a large corpus of English data and it has 109M parameters.

4 Experimental Setup

4.1 Preprocessing

We apply some preprocessing only on English data, when we use Machine Learning classifiers. There is no preprocessing on English data when we use

LLMs (either for embeddings or classification) or in the multilingual subtask.

The preprocessing is done with the following order. At first, we lowercase the text data and then we can either replace unicodes or remove them, but since we see that the latter has better results, we remove them. After that, we replace the emails with the word <email> and the URLs with the word <url>. We, also, remove the digits and all punctuations. Considering both tokenization and lemmatization, we see that tokenization performs better. Moreover, we achieve better results by the removal of stop words. Finally, we create some new features from the text data. These features are the number of times some phrases, words or combinations of punctuations ("cannot", "do not", "." and more) appear in each text, the number of characters on the original texts and the number of words after tokenization.

4.2 Embeddings

For SubtaskA monolingual, we try to get the embeddings using Word2Vec(Mikolov et al., 2013), Tf-Idf(Ramos et al., 2003) and BERT encoder. On Word2Vec we try the following vector sizes 20, 40, 60, 80, 100, 150, 200 and 300. On Tf-Idf we try the following X more frequent words 500 and 1000 with Ngrams of (3,5). Finally, on BERT encoder we get embeddings from the last of the 12 layers.

After comparing all the above, we see that the best results are coming from Word2Vec with vector size 20.

On SubtaskA multilingual, we get embeddings with the multilingualBERT encoder from its last layer.

We use Word2Vec with vector size 20 as on SubtaskA monolingual, to get embeddings on SubtaskB text data too, in order to see the performance of some ML algorithms on this task.

4.3 Machine Learning Algorithms VS LLMs

The metric we use is accuracy. We have seen that micro-f1 and macro-f1 values fluctuate according to the accuracy value. We, also, standardize the embeddings before we feed them into the ML algorithms. The accuracy values are calculated based on the preprocessing mentioned above except for the part of stop words. When stop words are removed it is specified on the table. Also, when we do not standardize the input data we mention it on the table.

In this section, all models are trained and evaluated using the training set and validation set the organizers give us. All values are calculated on the validation set. ML algorithms without * or additional information presented on the tables, have the default parameters of scikit-learn and XGBClassifier libraries (versions 1.3.0 and 1.7.3 respectively).

4.3.1 SubtaskA monolingual

We start to get embeddings using Word2Vec different vector sizes and compare them based on Logistic Regression. The results are on the Table 1.

We can see that Word2Vec embeddings with vector size 20 is the best based on LR. Now, we take the best embeddings, with vector size 20, and try different ML algorithms to see what results we can take on the validation set. The results are on the Table 2.

We can see that the best result here is default AdaBoost (Freund and Schapire, 1997) with optimized RandomForest (Breiman, 2001) and removed stop words. We have also tried optimizations to other algorithms and some voting ensembling methods with some of the best results from above, but everything was worse than the best one.

Now, we try different methods on getting embeddings to see if anything can beat Word2Vec with vector size 20 based on Logistic Regression and Random Forest. The results are on Table 3.

Now, we compare the best result from the ML algorithms with miniLM and BERT. Both miniLM and BERT are trained for 5 epochs on the training set and evaluated on the validation set. We evaluate them on every 100 batches, and we take the mean of all evaluations. The results are on the Table 4.

4.3.2 SubtaskA multilingual

We take embeddings using the last layer of multilingualBERT encoder and try some ML algorithms. The results are on the Table 5.

We can see from the algorithms compared that the best here is XG Boost(Chen and Guestrin, 2016) with no standardization applied.

Now, we compare the best ML algorithm with multilingualBERT and XLMRoBERTa as classifiers. The results are on Table 6 and again for the LLMs' values, because we evaluate them on every epoch from the 5, we take the mean of them.

We can see here that XLM-RoBERTa is slightly better from default XgBoost. So, this is the best model for this task.

	vector size						
Algorithm	**20**	**40**	**60**	**80**	**150**	**200**	**300**
Logistic Regression	**0.559**	0.553	0.5164	0.5046	0.4848	0.4854	0.4902

Table 1: Logistic Regression accuracy per vector size (Word2Vec).

Model	**Acccuracy**
Logistic Regression	0.559
XG Boost	0.7362
Decision Tree (Breiman, 2017)	0.6946
SGDC *	0.5532
Random Forest	0.7538
Random Forest optimized	0.7552
AdaBoost optimized	0.756
**AdaBoost ** **	**0.7584**
Bagging optimized (Breiman, 1996)	0.7538

Table 2: Machine Learning algorithms accuracy on monolingual validation set. *modified huber loss. ** Adaboost with optimized random Forest with removed stop words.

4.3.3 SubtaskB

On this subtask we take embeddings using the last layer of BERT and try some ML algorithms. The results of how ML algorithms perform on this multiclass task are presented on Table 7.

5 Modification on datasets

5.1 Rationale

We make some comparisons between datasets, and we decide to create new training and validation sets for subtaskA and subtaskB. We notice that subtaskA multilingual training set contains all the English data of the rest datasets and some extra, which means that it has the most English data. So, we decide to create a new monolingual dataset with all English data. We, also, notice that in the multilingual training set there are only the English, Chinese, Indonesian, Urdu and Bulgarian data and on multilingual validation set there are the three other languages. Thus, we create a new dataset with all multilingual data containing all languages. Now, from these two new datasets we create the new training and validation sets of subtaskA and subtaskB.

5.1.1 SubtaskA monolingual

Using the dataset with all the English data, we keep 131589 for training and 5000 for validation, where the 2500 texts are human-written, and the 2500 texts are machine-generated. With these new datasets we train and evaluate BERT. Because we want to train multilingualBERT, also, to see its performance on English data, we take the multilingual training set given from organizers and exclude the same as before 5000 English data for validation. By this way, this training set has the same remaining 131589 English data for training and the same 5000 English data for evaluation, with the difference now that this training set has 4 more languages (Chinese, Indonesian, Urdu, Bulgarian) and not only English data.

Both models are trained for 5 epochs and they are evaluated on all 5 epochs. The results are on Table 8 and both values are the mean of all their 5 evaluations.

5.1.2 SubtaskA multilingual

Using the dataset with all multilingual data, we create a new training set and a new validation set that contain texts from all languages. Specifically, the training set contains 133589 English, 10000 Chinese, 5000 Indonesian, 5000 Urdu, 10000 Bulgarian, 900 Arabic, 1800 Russian and 900 German data. The validation set contains 3000 English, 1934 Chinese, 995 Indonesian, 899 Urdu, 2000 Bulgarian, 100 Arabic, 200 Russian and 100 German with 50 percent human-written texts and 50 percent machine-generated texts. There is no specific technique behind the chosen percentages of each language.

We train the XLM-RoBERTa on this new training set for 5 epochs and make evaluations on each one of the 5 epochs. The result is 95.5 percent and it is the mean of all 5 evaluations.

5.1.3 SubtaskB

Using the dataset with all English data, the training and validation set for SubtaskB given from organizers, we concatenate the training and validation sets to compare them with the dataset of all English data. We can see that there are 62562 English data that are not used on this subtask. In these 62562 data there are different percentages of each class from the 6 (human, chatGPT, cohere, davinci, bloomz and dolly). We keep the same sample of

Algorithm	**W2V with 20 VS**	Tf-Idf 500 *	Tf-Idf 1000 *	Bert last layer
Logistic Regression	0.559	0.6288	0.6348	0.6618
Random Forest	**0.7552**	0.6226	0.7132	0.654

Table 3: Logistic Regression and Random Forest accuracy per different embedding methods. *With Ngram

Model	**Accuracy**
AdaBoost *	0.7584
MiniLM	0.7695
BERT	**0.783**

Table 4: Comparison of best machine learning algorithm with evaluated LLMs on monolingual task. *Adaboost with optimized Random Forest with removed stop words.

1844 texts from each class. The value is 1844, because in the 62562 texts, one of the classes has only this amount.

So, we create a new training set where we just add on the training set given from organizers these 1844 samples from each class. The validation set is the same. On this new training set we train BERT for 5 epochs, as we have seen on the other tasks that LLMs more often than not beat Machine Learning Algorithms. We evaluate BERT on every 10000 batches using the validation set organizers give us. The result is 96.81 percent and it is the mean of all evaluations.

5.2 Final Models

Finally, we combine on every subtask the new training set and validation set we created. We train the best models for 5 epochs on these datasets.

We can say that LLMs, most of the times, perform better on these tasks than Machine Learning algorithms. Nevertheless, there are some ML algorithms, combined with the right preprocessing and embeddings' method, that can give good results close to those LLMs give. As we can see, default AdaBoost with optimized Random Forest and with removed stop words achieves a quite good performance on subtaskA monolingual. The preprocessing made and the features we created on subtaskA monolingual seem to improve the performance of algorithms. Also, Default XgBoost with no standardization achieves a close enough to the best model performance on subtaskA multilingual. We believe that LLMs perform better because they are pre-trained models on a large corpus of English or multilingual data. Thus, they can better understand the meaning of a word and a whole text, and maybe this makes it easier for them to differentiate human-written from machine-generated texts.

Finally, the models submitted on the competition scored 73.243 on subtaskA monolingual, 76.45 on subtaskA multilingual and 56.683 on subtaskB. This means about 20 percent below for subtaskA and 30 percent below for subtaskB from the scores we had on the new validation sets we created from the dataset organizers give us. We think that this drop is due to the fact that the texts on the test sets are coming from different domain, generator and language. Basically, on subtaskA monolingual all texts are coming from a new domain "outfox" and there is also a new generator "gpt-4". On subtaskA multilingual, again the texts are coming from the same new domain and the languages it consists of are German, Arabic and Italian. The two first were also on the training but in a small amount and the 3rd one was not in the training set. On subtaskB, the only difference is the domain, which is the same as on every subtask, "outfox".

6 Conclusion

Based on the results we have on our new validation set and the results on the test set, we assume that this drop occurs since our model cannot generalize well. We believe that if the test sets had texts coming from the same domains and generators as the training texts, and had the same languages, then our models would have achieved better results.

Future work could focus on either training larger language models or trying to improve generalization of ours, possibly with some preprocessing like data augmentation. We look forward to further research on these tasks, hoping for better results.

References

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24:123–140.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Leo Breiman. 2017. *Classification and regression trees*. Routledge.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Diego de Vargas Feijó and Viviane Pereira Moreira. 2007. Mono vs multilingual transformer-based models: a comparison across several language tasks. *CoRR, abs*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Akim Tsvigun, Jinyan Su, Artem Shelmanov, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024c. MG-Bench: Evaluation benchmark for black-box machine-generated text detection.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

A Appendix

Model	Accuracy
Logistic Regression*	0.606
XGBoost*	**0.663**
Random Forest*	0.598
AdaBoost**	0.500
XGBoost	0.654

Table 5: Comparison of Machine Learning algorithms accuracy on multilingual task. * Without standardization. ** AdaBoost with XGBoost and without standardization.

Model	Accuracy
XGBoost*	0.663
Multilingual BERT	0.577
XLM-RoBERTa	**0.668**

Table 6: Comparison of best machine learning with evaluated LLMs on the multilingual. * Without standardization

Model	Accuracy
Logistic Regression	0.4636
XGBoost	**0.4720**
Gradient Boosting	0.4523

Table 7: Comparison of Logistic Regression, XGBoost and Gradient Boosting (Friedman, 2001) on sub-task B.

Model	Accuracy
BERT	0.9430
MultilingualBERT	**0.9540**

Table 8: Comparison of BERT and MultilingualBERT on the monolingual new validation set

Hyperparameter	Range/Value
Epochs	5
Batch size	16
Weight decay	0.02
Learning rate	2e-5

Table 9: Hyperparameter values for the multilingual BERT and XLM-RoBERTa.

Hyperparameter	Range/Value
Epochs	5
Batch size	16
Weight decay	0.03
Learning rate	2e-5

Table 10: Hyperparameter values for BERT.

IUST-NLPLAB at SemEval-2024 Task 7: Numeral Prediction using gpt3.5

Sina Alinejad
Iran University of
Science and Technology
sinaalinejad4@gmail.com

Erfan Moosavi Monazzah
Iran University of
Science and Technology
moosavi_m@comp.iust.ac.ir

Sauleh Eetemadi
Iran University of
Science and Technology
sauleh@iust.ac.ir

Abstract

In this paper, we present our approach to the SemEval-2024 numeral reasoning task, which requires filling in a blank with a number based on a given sentence. We first attempted to predict the arithmetic operation needed to compute the correct answer and obtained some statistical insights from this process. We performed operation prediction in two ways: as a 9-class classification problem and as a set of binary classification problems for each operation. However, due to the low accuracy of this method, we switched to a zero-shot learning strategy that leverages natural language inference models to solve the task.

1 Introduction

Headline generation is the task of summarizing a full-length article into a brief, catchy, and informative line of text. A key challenge in this task is to preserve the numerical information from the article, as numerals often convey important facts and figures. However, existing encoder-decoder models, despite achieving high ROUGE scores, tend to generate inaccurate or unreasonable numerals in headlines. One of the main reasons for this problem is the scarcity of datasets that provide detailed annotations for numeral generation.

To address this gap, the authors of (Huang et al., 2023) introduce the NumHG dataset, which consists of more than 27,000 numeral-rich news articles with fine-grained annotations. These annotations indicate how the numerals in the headlines can be derived from the numerals in the articles, using various arithmetic operations and transformations. The NumHG dataset enables the evaluation of numeral accuracy, reasonableness, and readability in headline generation. Moreover, the dataset covers both English and Chinese languages, allowing for cross-lingual studies. By emphasizing the role of numerals, the NumHG dataset aims to advance the state-of-the-art in number-focused headline generation and foster further research in numeral-focused text generation.

In this paper, we present our system for the NumHG task, which is based on zero-shot learning using gpt3.5. We first apply some preprocessing steps to the dataset, such as tokenization, normalization, and masking. Then, we use gpt3.5 to generate headlines by reformulating the task as a natural language inference problem. We compare our system's performance on different types of operations, such as copy, trans, paraphrase, round, subtract, add, span, divide, multiply, and sround. We find that our system performs well on some operations, such as copy and trans, but poorly on others, such as round. Our system ranks 12th in the leaderboard with an accuracy of 74 percent.

Additionally, we have made our code openly accessible on GitHub[1] to facilitate reproducibility and further research endeavors.

2 Background

2.1 Dataset Description

There are 21157 samples in the training set and the validation set contains 2572 samples. each sample contains the fields "news", "masked headline", "calculation" and "ans". Table 1 demonstrates an example from the dataset. The objective is to ensure accurate numeral generation in headlines, and as such, detailed annotations on how to secure the correct numeral through specific operations are provided. The whole dataset is in the English language. In this task, we are asked to predict the correct numeral value that the masked headline must be filled with based on the news.

[1] https://github.com/sinaalinejad/SemEval2024_task7_NumEval

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1087–1091

2.2 Related Work

The task of headline generation, a form of text summarization, endeavors to condense a lengthy source text into a succinct summary. Text summarization approaches typically fall into two categories: extractive and abstractive. Extractive approaches involve selecting fitting sentences from the source text to serve as the summary, while abstractive approaches strive to create new sentences to encapsulate the source text. The concept of headline generation aligns more closely with abstractive methodologies.

The emergence and development of large-scale pre-trained models like Lewis et al., Raffel et al. and Zhang et al., have notably advanced the capabilities of abstractive summarization models, to the extent that they now outperform extractive models. Some recent studies like Dou et al., Liu et al. and Wang et al., emphasize the significance of keyword sentences, asserting that these should be leveraged as guides for summary generation. GSum (Dou et al., 2021), for example, initially performs extractive summarization, then incorporates the extractive summaries into the input for abstractive summarization. Despite experimental evidence supporting GSum's effectiveness, Wang et al. argue that extractive summaries do not provide a reliable or flexible guide, potentially leading to information loss or noisy signals.

To tackle this issue, SEASON(Wang et al., 2022) adopts a dual approach, learning to predict the informativeness of each sentence and using this predicted information to guide abstractive summarization. Meanwhile, BRIO(Liu et al., 2022) employs pre-trained abstractive models to generate candidate summaries, assigning each a probability mass according to their quality and defining a contrastive loss across the candidates. By considering both token-level prediction accuracy and sequence-level coordination, BRIO combines cross-entropy loss and contrastive loss for abstractive summarization.

Table 1: An annotation example in NumHG.
News: At least 30 gunmen burst into a drug rehabilitation center in a Mexican border state capital and opened fire, killing 19 men and wounding four people, police said. Gunmen also killed 16 people in another drug-plagued northern city. The killings in Chihuahua city and in Ciudad Madero marked one of the bloodiest weeks ever in Mexico and came just weeks after authorities discovered 55 bodies in an abandoned silver mine, presumably victims of the country's drug violence. More than 60 people have died in mass shootings at rehab clinics in a little less than two years. Police have said two of Mexico's six major drug cartels are exploiting the centers to recruit hit men and drug smugglers, ...
Headline (Question): Mexico Gunmen Kill _____
Answer: 35
Annotation: Add(19,16)

Table 1: An annotation example in NumHG.

3 System Overview

3.1 Zero-Shot system

Our system is simply inferring the output by zero-shot learning. The input is given to gpt-3.5-turbo along with a prompt. The prompt is: "Act as a news expert. I have a text of news and its masked headline with a mask token specified as [MASK]. The mask should be filled with a numerical value. you should just give me the numerical value to put instead of the [MASK]. You should do some calculations to obtain the final number to put instead of [MASK] and these calculations are as follows: Copy(v): Copy v from the article Trans(e): Convert e into a number Paraphrase(v,n): Paraphrase the form of digits to other representations Round(v,c): Hold c digits after the decimal point of v Subtract(v0,v1): Subtract v1 from v0 Add(v0,v1): Add v0 and v1 Span(s): Select a span from the news Divide(v0,v1): Divide v0 by v1 Multiply(v0,v1) Multiply v0 and v1 the news is: <NEWS> the masked headline is: <MASKED HEADLINE>. your response should be in the format of JSON with the key of ans and value of the numerical answer, so do not include any of your calculation processes."

In the next step, we tried this system on a set of 100 samples from the training dataset with different prompts and the best result was an accuracy of 80 percent. Then we decided to have a set of 200 samples from the validation set but this time, the distribution of different records based on the field "calculation" was the same as the whole validation set; This time the accuracy was 77 percent.

At the end, we extract the number from the model response.

Metric	Value
Precision	0.32
Recall	0.27
F1	0.28
Accuracy	0.81

Table 2: Different metrics in operation prediction in 9-way classification using gpt2

Operation	Acc	Operation	Acc
Copy	0.9	Add	0
Trans	0.64	Span	0
Paraphrase	0.72	Divide	0
Round	0.37	Multiply	0
Subtract	0.05	Sround	0

Table 3: Accuracies for each operation in operation prediction in 9-way classification using gpt2

3.2 Operation prediction

In our investigation, we endeavored to forecast arithmetic operations using textual information extracted from news articles. To achieve this, we meticulously fine-tuned the GPT-2 language model for this specific task. The culmination of our efforts yielded the following outcomes:

Model Fine-Tuning: We conducted rigorous fine-tuning of the GPT-2 model, adapting it to the novel context of arithmetic prediction based on news content.

Performance Evaluation: Subsequently, we evaluated the model's accuracy for each arithmetic operation. The results are shown in Table 2:

The results for each operation are succinctly summarized in Table 3.

In our research endeavor, we revisited the application of the GPT-2 language model to binary classification tasks. Specifically, we aimed to predict the outcome of various arithmetic operations. Our investigation involved meticulous dataset creation, model fine-tuning, and performance evaluation. Below, we outline the key steps and findings of our study. To construct robust binary classification datasets, we adhered to a balanced approach. For each arithmetic operation (e.g., addition, subtraction, multiplication, etc.), we meticulously curated positive and negative samples.

1. Positive Samples: We collected all positive samples corresponding to each arithmetic operation.

2. Negative Samples: Achieving parity between positive and negative samples was crucial. Therefore, we ensured that the number of negative samples matched that of positive ones. However, the challenge lay in diversifying the negative samples. To address this, we introduced Distribution-Based Sampling in which each arithmetic operation in the negative sample was selected based on its distribution across all negative instances. For instance, if we were dealing with the "copy" operation and we gathered 50 positive instances from relevant data sources and the "trans" operation constituted 10% of all negative samples, we allocated 5 negative samples specifically for this operation.

$$50 * 0.1 = 5$$

The results for this method was around random classification, so we didn't continue on that.

We provide both the 9-way classification dataset and the binary classification dataset on HuggingFace[2] for public use. Researchers and practitioners can leverage these datasets for future investigations.

Our study underscores the challenges in predicting arithmetic outcomes from news content. Future research could explore alternative models, feature engineering techniques, or domain-specific adaptations to enhance classification accuracy. Additionally, investigating the impact of dataset size and quality on model performance remains an open avenue for exploration.

In summary, while our initial results did not yield groundbreaking accuracy, the datasets we present serve as valuable resources for the scientific community. As the field of natural language processing continues to evolve, we remain optimistic about refining predictive models for diverse applications.

4 Experimental Setup

4.1 Pre-processing

The news and the masked headline are pre-processed in these manners:

1. converting new line character and tab to space

2. removing the commas from comma-separated numbers, this can help the model to better understand the numbers

[2] https://huggingface.co/Sina-Alinejad-2002

3. replacing the blank in the masked headline with a new mask

4. converting some unknown characters to the closest ASCII equivalent for example uff05 to %. This makes the context easier to understand for the model

4.2 Evaluation Metrics

As this is a prediction task, we have used an accuracy metric to evaluate our model. However, there is no training stage in our system, so this metric is not used to update any parameter and it is just for us to change some hyperparameters like the prompt.

4.3 Others

We also used the tenacity library to handle some errors that may cause the cell to stop such as TimeLimit error or RateLimit error. For this, we set a retry decorator for the main function wait for 20 seconds after an error has occurred, and retry the request to API and this is for a maximum of 3 times.

```
@retry(stop=stop\_after\_attempt(3),
wait=wait\_fixed(20))
```

5 Results

5.1 Overall Performance

The overall performance of our system on the test dataset was 74 percent. We also calculated the accuracy of each of the 10 operations and the result is shown in table 4.

5.2 Error Analysis

The system performs poorly on predicting answers that require the round operation to be applied and this is probably because the model tends just to copy the exact number in the blank or round it in different ways. On copy and trans operations, the results are the best compared to others which are around 50 percent.

6 Conclusion

In this paper, we have presented a zero-shot learning system for the NumHG task, which leverages gpt3.5 to generate headlines with accurate and reasonable numerals. Our experimental results show that our system can handle simple operations, such as copy and trans, but fails to perform complex operations, such as add, subtract, and round. This indicates that current LLMs like gpt3.5 still have limitations in capturing the numerical reasoning and arithmetic skills required for the NumHG task.

Operation	Acc	Operation	Acc
Copy	0.82	Add	0.46
Trans	0.81	Span	0.5
Paraphrase	0.54	Divide	0.54
Round	0.02	Multiply	0.4
Subtract	0.5	Sround	0

Table 4: Accuracies based on the operation used to calculate the answer

For future work, we propose to explore the possibility of using multiple agents to collaborate on the task. This could involve either having different agents specialize in different operations or having a voting mechanism to select the best answer from multiple agents. We believe that this could improve the overall performance and robustness of our system, and also provide more insights into the strengths and weaknesses of different LLMs.

We suppose that it would be also useful to extend the exploration of numeral reasoning tasks by incorporating few-shot learning techniques. This approach will allow us to delve deeper into the performance enhancements across various operations, providing a more granular understanding of the model's capabilities. Furthermore, we can transcend beyond merely predicting the final answer. Inspired by the iterative prompting methodology of Chain of Thought (Wei et al., 2023), it would be possible to endeavor to refine our model's reasoning process. This will involve guiding the model to deduce the correct set of operands and the associated operation before executing it, thereby fostering a more transparent and interpretable reasoning pathway. Such advancements will not only bolster the model's accuracy but also its ability to articulate the reasoning behind its conclusions, paving the way for more robust and reliable numeral reasoning systems. The impact of dataset size and quality is also an open avenue to explore, one such experiment has been conducted by (Jain et al., 2020).

References

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang,

and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. 2020. Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3561–3562, New York, NY, USA. Association for Computing Machinery.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Fei Wang, Kaiqiang Song, Hongming Zhang, Lifeng Jin, Sangwoo Cho, Wenlin Yao, Xiaoyang Wang, Muhao Chen, and Dong Yu. 2022. Salience allocation as guidance for abstractive summarization.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

IUSTNLPLAB at SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes

Mohammad Osoolian
Iran University of Science and Technology
dsoolian@gmail.com

Erfan Moosavi Monazzah
Iran University of Science and Technology
moosavi_m@comp.iust.ac.ir

Sauleh Eetemadi
Iran University of Science and Technology
sauleh@iust.ac.ir

Abstract

This paper outlines our approach to SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes, specifically addressing subtask 1 in English language. The study focuses on model fine-tuning using language models, including BERT, GPT-2, and RoBERTa, with the experiment results demonstrating optimal performance with GPT-2. Our system submission achieved a competitive ranking of 17th out of 33 teams in subtask 1, showcasing the effectiveness of the employed methodology in the context of persuasive technique identification within meme texts.

1 Introduction

Propaganda is the term used when information is intentionally molded to promote a specific agenda. Memes typically involve combining an image with text. In deceptive memes, the image serves to either enhance or complement a technique employed in the text, or it independently conveys one or more persuasive techniques. In subtask 1 of SemEval-2024 Task 4 (Dimitrov et al., 2024), the challenge involves identifying which of the 20 persuasion techniques, organized hierarchically, are utilized, based on the textual content of a meme.

For this problem, GPT2 was chosen as the base model after experiments on GPT2, BERT and RoBERTa. After that, the model was fine-tuned on the given data set and after doing error analysis and comparing them with true labels, the threshold of sensitivity was changed manually to get best results. In addition, we tried to fine-tune model on SemEval-2023 Task 3 dataset which is similar to the given dataset for this task. However, the results didn't improve.

Regarding the noticeable change in scores just by changing the threshold of predicting labels, we realized the importance of error analysis and the easy tricks comes after actually understanding the behavior of model and it's problems.

We have made all the code necessary to replicate our results available in the paper's GitHub repository.[1]

2 Background

2.1 Dataset Description

The dataset consists of 7000 samples for training and 500 samples for validation. each sample contains three fields:

- id: A unique identifier assigned to each sample, facilitating the association with the corresponding meme image. It is noteworthy that, for the purposes of Subtask 1, the visual components of the memes, indicated by these IDs, are not considered in the training.
- text: this field is the textual content of the meme, as a single UTF-8 string. While the text is first extracted automatically from the meme, it has been post-processed to remove errors and formated in such a way that each sentence is on a single row and blocks of text in different areas of the image are separated by a blank row.
- label: it is a list of valid technique names used in the text. There are 22 techniques in this dataset which are leaf nodes of the hierarchy of persuasion techniques shown in Figure 1. However, only 20 of them are used for subtask 1.
- link: This field contains the social network link associated with the meme. It is imperative to acknowledge that certain samples may lack a corresponding link. In such cases, the term "null" is employed in lieu of a link.

You can see an example of training samples in Figure 2.

[1] https://github.com/mohammad-osoolian/SemEval-2024_task4

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1092–1096

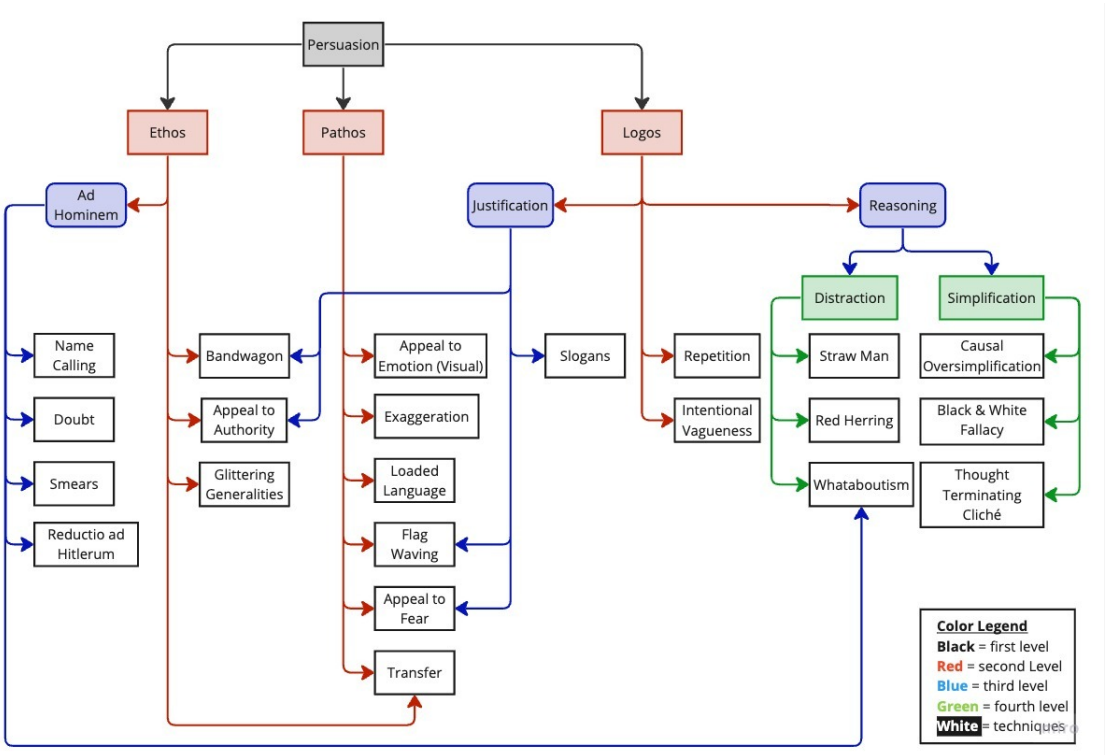


Figure 1: Hierarchy of persuasion techniques (Dimitrov et al., 2024)


```
{
"id": "66730",
"text": "WHEN THE POWER OF LOVE IS GREATER THAN THE
         LOVE OF POWER, THE WORLD WILL KNOW PEACE",
"labels": [
           "Loaded Language",
           "Black-and-white Fallacy/Dictatorship",
           "Slogans"
           ],
"link": "null"
},
```


Figure 2: An example in the training set

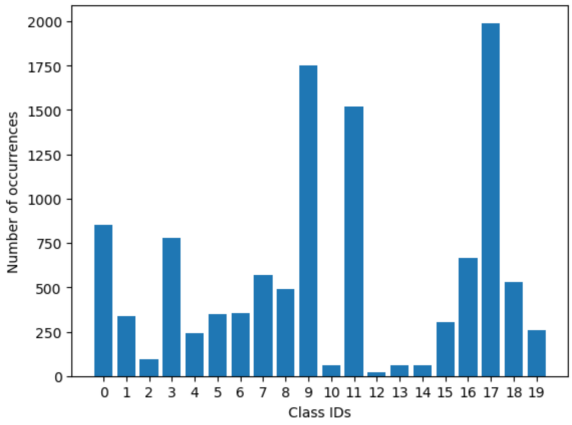


Figure 3: Number of Class occurrences in the labels

The distribution of classes in train data is shown in the Figure 3 and class names are shown in the Table 1.

2.2 Related Works

Prior to the SemEval 2024 event task, researchers have endeavored to address analogous challenges, contributing to the evolution of methodologies for detecting persuasive techniques in multimodal content.

The article "Detecting Propaganda Techniques in Memes" (Dimitrov et al., 2021) establishes a novel multi-label, multimodal task of automatically detecting propaganda techniques in memes. creating a dataset of 950 annotated memes covering 22 propaganda techniques, the authors provide a crucial resource for training and evaluating future detection models. In addition, by creating a dataset of 950 annotated memes covering 22 propaganda techniques, the authors provide a crucial resource for training and evaluating future detection models.

The article "SemEval-2023 Task 3: Detecting the Category, the Framing, and the Persuasion Techniques in Online News in a Multi-lingual Setup" (Piskorski et al., 2023) provides a publicly available dataset of annotated news articles, along with code and evaluation metrics. These resources serve as a valuable starting point for future research and development in multilingual news analysis tasks.

Number	Fallacy
0	Appeal to authority
1	Appeal to fear/prejudice
2	Bandwagon
3	Black-and-white Fallacy/Dictatorship
4	Causal Oversimplification
5	Doubt
6	Exaggeration/Minimisation
7	Flag-waving
8	Glittering generalities (Virtue)
9	Loaded Language
10	Misrepresentation of Someone's Position (Straw Man)
11	Name calling/Labeling
12	Obfuscation, Intentional vagueness, Confusion
13	Presenting Irrelevant Data (Red Herring)
14	Reductio ad hitlerum
15	Repetition
16	Slogans
17	Smears
18	Thought-terminating cliché
19	Whataboutism

Table 1: Class names and their IDs

2.3 Task evaluation and ranking

The hierarchical taxonomy of labels in this task necessitates a nuanced approach to evaluation. According to the task description, when predicting the ancestor node of a technique, only a partial reward is assigned, highlighting the hierarchical multilabel classification nature of the problem at hand.

To assess the performance of submissions, the chosen metric is the hierarchical F1 score (Kiritchenko et al., 2006). Hierarchical f1 score is a way of adapting the F1 score metric to be used for classification tasks with hierarchical structures. These structures involve classes having parent-child relationships, forming a kind of tree-like organization. It is crucial to note that the conventional F1 score is designed for flat classifications devoid of hierarchical relationships, making the hierarchical F1 score a pertinent choice for the evaluation of this task.

3 System overview

3.1 Model Architecture

Initially, our approach involved the utilization of three distinct models: BERT, RoBERTa, and GPT-2, all of which were subjected to fine-tuning on the training set. Subsequent evaluation based on the metrics outlined earlier revealed that the performance of the GPT-2 model surpassed that of its counterparts. (Table 2)

Given the superior performance observed with the GPT-2 model, we proceeded with this architecture for further refinement. The fine-tuning process ensued, culminating in the generation of our final results, which were subsequently submitted utilizing the GPT-2 model.

3.2 Fine tuning on extra dataset

Following the initial training on the provided dataset, our exploration extended to leveraging comparable datasets from previous studies and SemEval events. The SemEval-2023 Task 3 dataset, encompassing paragraphs extracted from diverse news articles and publications annotated with 19 distinct propaganda techniques, emerged as a pertinent source for augmenting our training data.

To ensure compatibility and coherence between the SemEval-2023 Task 3 dataset and our specific task dataset, a meticulous data cleaning process was undertaken. This involved the removal of uncommon tags, resulting in a curated dataset comprising 3,445 new samples. This augmented dataset was then incorporated into the fine-tuning phase of our model, aiming to enhance its adaptability and robustness across diverse text corpora.

3.3 Adjusting the prediction threshold

The model generates continuous probability values reflecting the likelihood of the presence of various persuasion techniques within the input text, rather than providing explicit binary predictions. A threshold is applied to discretize these probability values, where a value exceeding the threshold results in a prediction of 1, and otherwise, it is predicted as 0. The adjustment of this threshold played a pivotal role in refining the model's output, leading to noticeable improvements in performance.

In fine-tuning the threshold value, we tested a range of thresholds on the training set and assessed their performance using the F1-score. Based on the results depicted in Figure 4, we settled on a threshold value of 0.19.

Model	Accuracy	Precision Macro AVG	Recall Macro AVG	F1-Score Macro AVG
BERT	0.218	0.403	0.201	0.238
RoBERTa	0.232	0.344	0.179	0.2145
GPT2-medium	0.382	0.637	0.423	0.489

Table 2: Evaluation Metrics for GPT2, BERT and RoBERTa models on validation set

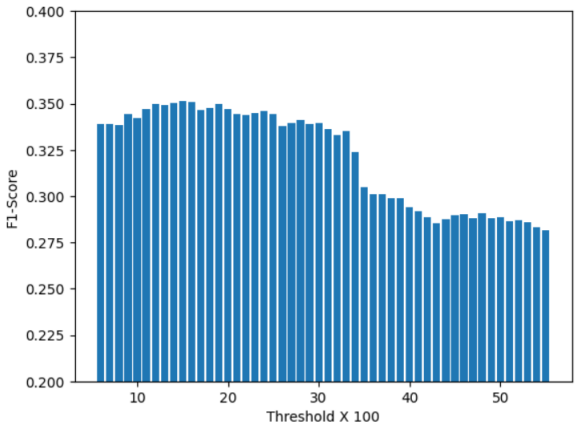


Figure 4: Adjusting the prediction threshold with F1-Score

4 Experimental setup

4.1 Dataset Split

The dataset is partitioned into distinct sets to facilitate comprehensive training, validation, and evaluation processes. The training set comprises 7,000 samples, while the validation set consists of 500 samples. The test set, used for submitting predictions, encompasses 1,500 samples.

Additionally, there exists a dev set that initially lacked labels and was subsequently annotated. This set was not incorporated into the model training process but only used to measure improvements in results and scores.

For the purpose of augmenting the training data, an extra dataset derived from SemEval-2023 Task 3 was considered. Following data cleaning, this supplementary dataset yielded 3,445 samples. However, despite this effort, training the model with the extra dataset did not yield discernible improvements. Consequently, the submitted model was not fine-tuned using this additional dataset.

4.2 Preprocessing Dataset

In the preprocessing of the main dataset, the initial step involved converting the data from JSON format to a tab-separated values (tsv) format. During this transformation, the "link" field in the samples was removed. The resulting dataset comprises columns for ID, Label, and Text.

As for the extra dataset, the samples were initially distributed across various files as paragraphs, with labels stored separately in different files. To align with the structure of the main dataset, each file was processed by splitting it into individual paragraphs. Subsequently, labels were gathered, and each paragraph was transformed into a unified sample with an assigned ID. To ensure compatibility, samples with labels not present in the main dataset were excluded, streamlining the integration of the extra dataset into the training process.

4.3 Evaluation Metrics

The system we have designed, only predicts the actual 20 classes which are the leaf nodes in the hierarchy of persuasion techniques. Therefore we have not used proposed hierarchical f1 score. The metrics we have used for our own evaluations are as follows:

- precision: Calculated individually for each class and expressed as total precision with macro averaging between classes. Precision serves to measure the accuracy of positive predictions.
- recall: Computed for each class and represented as total recall with macro averaging between classes. Recall measures the completeness of positive predictions.
- f1-score: Determined for each class and presented as the total F1 score with macro averaging. The F1-score serves as a comprehensive metric in classification tasks, considering both precision and recall.

5 Results

5.1 Overall Performance

Finally our model reached hierarchical f1-score of 0.624 and hierarchical precision of 0.631 and hierarchical recall of 0.617 in English language. In comparison, the baseline metrics for these categories were significantly lower at 0.368, 0.477, and 0.300, respectively. (Table 3)

Model	Hierarchical F1-score	Hierarchical Precision	Hierarchical Recall
First team	0.752	0.684	0.835
Our model (17th team)	0.624	0.631	0.617
Baseline	0.368	0.477	0.300

Table 3: Team ranking and model hierarchical scores

5.2 Analysis model predictions

By comparing the obtained results with the results of the first team, we see that the precision values are not much different, but the recall value for the first group is much higher than the recall of our model. This disparity indicates that our model may lack sensitivity and we can achieve better results by focusing on improving recall in the model.

6 Conclusion

In this paper, we examined different models and finally by choosing GPT2, we presented a model for the problem of identifying persuasion techniques in English memes. With the help of the presented model and adjusting the threshold for this model, we were able to reach a score of 0.624 for f1-score. Our work demonstrates the effect of choosing the appropriate model for training and the need to perform error analysis to improve the accuracy of the model.

References

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Detecting propaganda techniques in memes.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A. Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence*, pages 395–406, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

PWEITINLP at SemEval-2024 Task 3: Two Step Emotion Cause Analysis

Sofiia Levchenko[1], **01155482@pw.edu.pl**
Rafał Wolert[1], **01151705@pw.edu.pl**
Piotr Andruszkiewicz[1,2], **piotr.andruszkiewicz@pw.edu.pl**
[1]Warsaw University of Technology
[2]Samsung Research Poland

Abstract

ECPE (emotion cause pair extraction) task was introduced to solve the shortcomings of ECE (emotion cause extraction). Models with sequential data processing abilities or complex architecture can be utilized to solve this task. Our contribution to solving **Subtask 1: Textual Emotion-Cause Pair Extraction in Conversations** defined in the **SemEval-2024 Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations** is to create a two-step solution to the ECPE task utilizing GPT-3 for emotion classification and SpanBERT for extracting the cause utterances.

1 Introduction

This paper introduces an approach for the emotion-cause extraction problem in dialogues. An emotion cause is defined and annotated in the given subtask as a textual span. Input to the model is a conversation containing the speaker and the text of each utterance. The model output should include all emotion-cause pairs, where each pair contains an emotion utterance along with its emotion category and the textual cause span in a specific cause utterance, e.g.(U3_Joy, U2_"You made up!").

Our contribution to solving **Subtask 1: Textual Emotion-Cause Pair Extraction in Conversations** defined in the **SemEval-2024 Task 3: The Competition of Multimodal Emotion Cause Analysis in Conversations** (Wang et al., 2024) is as follows: i) utilize GPT-3 for emotion classification, ii) utilize SpanBERT architecture to extract the cause utterances in dialogues as Q&A task, iii) contribute to solving the ECPE by finding its possible solutions in other NLP fields.

The task was separated into two parts - emotion classification, called subtask 1.1, and emotion-cause pair extraction, termed subtask 1.2. We have used GPT-3 and the SpanBERT model for these subtasks. In this paper, we also reflect on the results we got in the case of both subtasks, namely emotion classification and emotion-cause pair extraction. Our model, with one test entry, scored 9th in the competition.[1]

2 Related work

In recent years, many authors have suggested their approach to solving the ECPE task. Xia and Ding (2019) defined the ECPE task and proposed a two-step framework. First, independent multi-task learning (named Indep) consisting of BiLSTM modules and interactive multi-task learning (called Inter-EC for a model that uses emotion extraction to improve cause extraction and Inter-CE for a model that uses cause extraction to enhance emotion extraction) was used to extract a set of emotion cases and a set of cause clauses. Secondly, the sets were paired to yield a set of candidate emotion-cause pairs. Finally, a logistic model regression was used to filter the pairs. This two-step framework suffers from error propagation from the first step to the second step. Ding et al. (2020a) has also proposed a one-step framework that takes emotion-cause pairs as a 2D representation scheme with BiLSTM modules. These representations are forwarded into the 2D Transformer framework to capture pair interaction. Finally, binary classification is conducted to extract valid emotion-cause pairs. The new proposed framework outperforms the two-step framework by 7.6 percentage points of the F1 score. Regarding the joint framework, Ding et al. (2020b) have proposed a sliding window multi-label learning scheme named ECPE-MLL. It works on the assumption that all clauses in a document are emotion clauses, and an emotion-oriented sliding window is built centered on each emotion clause. In each window, the emotion clause extracts one or more of the corresponding cause clauses (the iterative synchronized multi-task learning (ISML) model is introduced to solve these subtasks). The results of this

[1]https://codalab.lisn.upsaclay.fr/competitions/16141#results

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1097–1105

learning can be transformed into emotion-cause pairs. This approach serves an excellent advantage over the two-step framework proposed before. Chen et al. (2022) have proposed two alignment mechanisms with a model named A^2Net. Text documents are encoded with BERT and a partition filter network (PFN) to implement the first alignment mechanism: feature-task alignment to produce emotion-specific, cause-specific, and interaction features. The features are applied for EE (emotion and interaction features), CE (cause and interaction features), and ECPE tasks (all features). The inter-task alignment reduces then the inconsistency between label spaces among all tasks. The proposed framework achieves a higher F1 score and recall in the ECPE task, a higher F1 score in the EE task, and a higher recall and F1 score in terms of the CE task when compared to ECPE-2D.

3 Methodology

The emotion extraction cause task consists of two components - emotion extraction from the conversation and emotion cause span extraction. The first one could have been considered as a baseline for the second one, as we needed to identify which emotion and utterance should be used in the process of the cause search. There are two ways of approaching this problem. We could have created one model for both subtasks or separated it into two subsequent tasks, where each could be implemented using different models.

We have decided to go with the second approach, as we concluded that those less complex parts could have better quality in the end, even though we are aware of the error propagation, which definitely will be present in such a case.

3.1 Subtask 1.1

The first subtask aims to create a classification model, which will perform emotion recognition in each utterance of the conversation.

We have focused on two different models while approaching this problem. At first, we decided to use BiLSTM, but the results were not promising (refer to Appendix A and Section 4.1.2). Then, we have focused on utilizing the GPT-3 model (Brown et al., 2020) along with AssistantAPI provided by OpenAI.

3.1.1 Dataset

The training dataset, presented by the SemEval competition organizers, contained information about conversations between groups of people and emotion-cause pairs extracted from that conversation. The conversation consisted of multiple utterances, each with defined text, speaker, and emotion expressed by the speaker and their id.

The given dataset was transformed into a set of objects, where each represents a single utterance along with information about the context (concatenated utterances within the conversation) and expressed emotion.

3.1.2 Model

For the GPT-3 model, we have decided to use a standard Assistant (by only defining its purpose) and one enriched with data retrieval (by adding properly labeled data as its knowledge base).

The purpose of both Assistants was defined using the description:

*You are a system which analyzes conversation which consists of utterances sequence (attribute "context" in the given JSON object) among with given utterance (attribute "utterance" in the given JSON object) and then predicts emotion expressed (fear, surprise, joy, disgust, sadness, anger or neutral, you cannot use any other emotion as an answer and you must detect at least one of those emotions) in that utterance adding it to the answer using "**" symbol to emphasize answer's location.*

The enrichment of the second Assistant was based on the OpenAI *Knowledge Retrieval* functionality [2]. A selected number of records described further in Section 4.1 were fed into the GPT-3 model as a knowledge base. Upon querying, the model performs either a vector search or passes the file content to the context of the model calls. For further clearance, a model with/without a knowledge base will be called *GPT-3 based Assistant with/without knowledge base*.

3.2 Subtask 1.2

The second subtask aims to find which utterances in a given context are responsible for inducing the emotion predicted in subtask 1.1 (for details please refer to Section 3.1). The main idea of the second subtask is to fine-tune the SpanBERT model (Joshi et al., 2020) and perform question-answering to find the utterances in the dialogue for predicted emotion.

[2] https://platform.openai.com/docs/assistants/tools/knowledge-retrieval

Dataset	Count
train	5635
validation	1349
test	2380

Table 1: Train, validation, test dataset sizes for subtask 1.2

3.2.1 Dataset

The raw dataset that was presented in Section 3.1 and split into the 0.6-0.15-0.25 ratio was transformed to fit the question-answering task. The duplicates were removed. The original **text** field combined with the person speaking **speaker** in each **conversation** in the provided SemEval (Wang et al., 2023) dataset was converted into the **context** column. The question to be answered was formulated as follows: *What caused the [emotion]?*, where *[emotion]* refers to the predicted emotion for a given utterance combined into the context. Additionally, information was provided to indicate where the answer starts in the context, and **text** column to show the answer in the context. Table 1 shows the dataset sizes used for subtask 1.2.

A tokenizer was used with the original SpanBERT to fit the dataset into the SpanBERT input. Along the tokenization process, the following preprocessing steps were also applied:

1. For questions (column **question**) and contexts (column **contexts**), tokenization with truncation and padding on the right was applied. The max length of sequences was set to 512 (default SpanBERT value), and the stride was also used and set to 128 so that if the context is long, each of the features retrieved from the context has a context that overlaps the context from the previous feature.

2. For answers, the start position and end position were marked so that the current span's token index and the current span's end token index were put correctly even if the answer is out of span (CLS token was added in that case). If the answer was in a given span, the token start index and token end index were put to the two ends of the answer.

3.2.2 Model

The model used to finetune the data prepared for subtask 1.2 was SpanBERT[3] (Joshi et al., 2020) finetuned previously on the SQuAD v1.1 for the Q&A downstream task.

[3]HuggingFace implementation has been used.

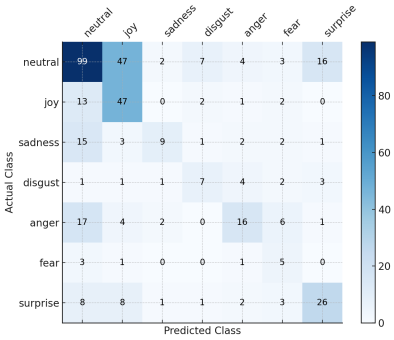


Figure 1: Confusion matrix for the GPT-3 based Assistant with knowledge base

4 Experimental Results

This section presents the experiment results (before evaluation phase) in terms of both subtasks.

4.1 Subtask 1.1

4.1.1 GPT-3 model

We have checked the accuracy of the GPT-3 model by utilizing 400 randomly selected utterances. Assistant, which was enhanced by adding a knowledge base, was using another randomly selected (but not similar to the ones in the test dataset) 500 records from the training dataset.

During the testing phase, we calculated the predicted labels' F1-score, accuracy, and recall and created a confusion matrix.

Figures 1, 2 and Tables 2, 3 refer to the confusion matrix for the Assistant with and without knowledge base accordingly.

Emotion	Precision	Recall	F1-Score
neutral	0.63	0.56	0.59
joy	0.42	0.72	0.53
sadness	0.60	0.27	0.37
disgust	0.39	0.37	0.38
anger	0.53	0.35	0.42
fear	0.22	0.50	0.30
surprise	0.55	0.53	0.54
Accuracy			0.52
Macro Avg	0.48	0.47	0.45
Weighted Avg	0.55	0.52	0.52

Table 2: Classification Report for the GPT-3 based Assistant with knowledge base

Emotion	Precision	Recall	F1-Score
neutral	0.62	0.26	0.37
joy	0.31	0.86	0.46
sadness	0.27	0.09	0.14
disgust	0.32	0.37	0.34
anger	0.43	0.48	0.45
fear	0.25	0.50	0.33
surprise	0.41	0.35	0.38
Accuracy			0.39
Macro Avg	0.37	0.42	0.35
Weighted Avg	0.47	0.39	0.37

Table 3: Classification Report for the GPT-3 based Assistant without knowledge base

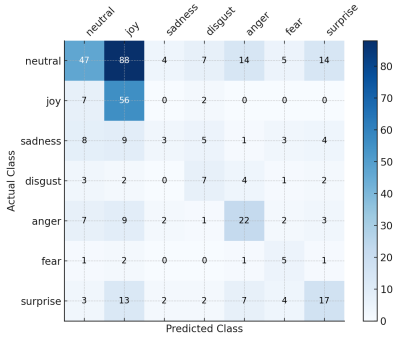


Figure 2: Confusion matrix for the GPT-3 based Assistant without knowledge base

By looking at the Tables 2, 3, one can see that average Macro and Weighted metrics are higher in all given cases: F1-score, Precision, and Recall when dealing with GPT-3 Assistant with knowledge base. Metrics for emotions such as *sadness* for GPT-3 based Assistant without knowledge are relatively low compared to much better results in terms of metrics when dealing with GPT-3 based Assistant with knowledge base. Figures 1 and 2 present the confusion matrices for two version of GPT-3 classifier. For GPT-3 based Assistant with the knowledge base, more *neutral* cases were predicted correctly. In contrast, without the knowledge base, more *neutral* cases were predicted incorrectly as *joy* class.

4.1.2 BiLSTM model

The following Figures 3, 4 and Tables 4, 5 refer to the confusion matrix for Model 1 and Model 2 accordingly used in the BiLSTM experiment (please refer to Appendix A for training and model details).

While analyzing the confusion matrix and values of the metrics for the test data, one can see that

Emotion	Precision	Recall	F1-Score
neutral	0.45	0.39	0.42
joy	0.24	0.35	0.28
sadness	0.14	0.09	0.11
disgust	0.06	0.03	0.04
anger	0.09	0.06	0.07
fear	0.05	0.04	0.04
surprise	0.20	0.32	0.25
Accuracy			0.28
Macro Avg	0.18	0.18	0.17
Weighted Avg	0.29	0.28	0.28

Table 4: Classification Report for Model 1

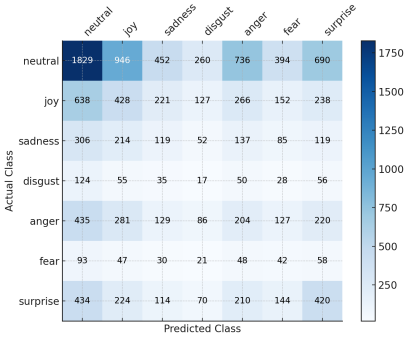


Figure 3: Confusion matrix for the Model 1

the results are not the best - weighted accuracy is around 0.3, while recall and F1-scores are approximately 0.28. Results for both Models are pretty similar, so we can only say that context was not utilized by us well enough for it to affect prediction results (Figures 3 and 4).

The performance of the model was also affected by the distribution of the labels (refer to Figure 5) - such an unbalanced dataset caused labels for neutral, joy, surprise, anger (and also sadness) were more likely to be classified in the right way than

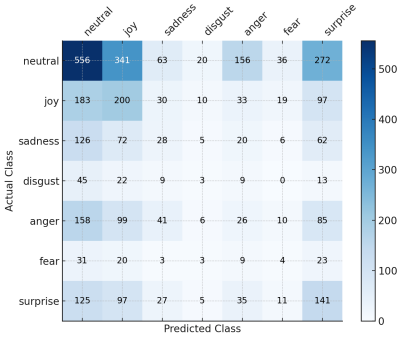


Figure 4: Confusion matrix for the Model 2

Emotion	Precision	Recall	F1-Score
neutral	0.47	0.34	0.40
joy	0.19	0.21	0.20
sadness	0.11	0.12	0.11
disgust	0.03	0.05	0.03
anger	0.12	0.14	0.13
fear	0.04	0.12	0.06
surprise	0.23	0.26	0.25
Accuracy			0.25
Macro Avg	0.17	0.18	0.17
Weighted Avg	0.30	0.25	0.27

Table 5: Classification Report for the Model 2

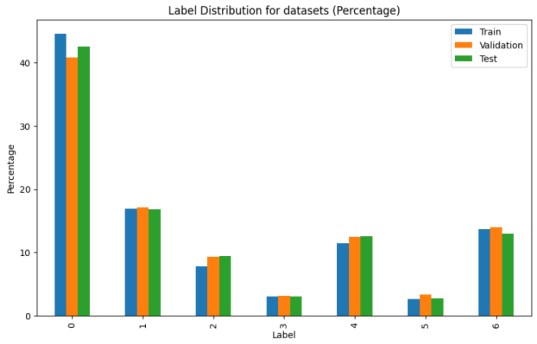


Figure 5: Label distribution in the dataset

fear and disgust ones.

When considering the best GPT-3 model with the knowledge base and all BiLSTM models, the GPT-3 overperforms the BiLSTM model in all presented metrics. It became clear that if we want to use BiLSTM models for the classification tasks where context plays an important role, there should be more complex preprocessing techniques and feature extraction for both the model input and the attention layer (both utterances and context values) that would be solved by utilizing GPT-3 model.

4.2 Subtask 1.2

Regarding fine-tuning the SpanBERT model, training and validation loss were calculated on the given dataset.

Two metrics were chosen to test the SpanBERT model. First is defined as **EM** or **exact match** and is defined as a sum of all of the individual exact match scores in the set, divided by the total number of predictions in the set. Also, the F1-score was used.

Table 6 summarizes the training configuration. The parameters were set so that the **learning rate** is minimized, **batch size** does not exceed the given

Parameter	Value
Learning rate	1e-5
Batch size	8
Training epochs	4
Weight decay	0.01

Table 6: Training config for subtask 1.2

Metric	Value
EM	21.42
F1-score	33.87

Table 7: Exact match and F1-score for Q&A task

RAM of the machine, **training epochs** was set to between 2 and 4 according to BERT's authors' (Devlin et al., 2019) and **weight decay** was set to default.

Figure 6 presents the training and validation loss. The scores obtained for the Q&A task on the test

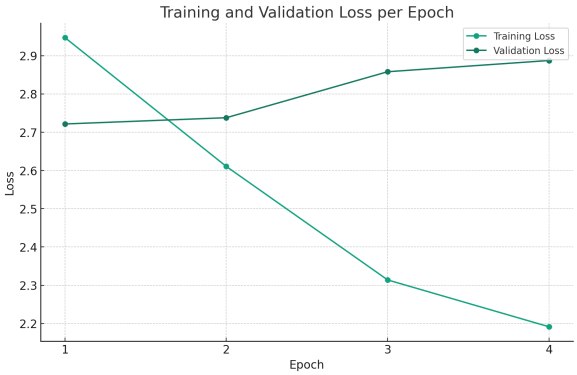


Figure 6: Training and validation in the epochs

dataset are shown in Table 7.

4.3 Results Analysis

4.3.1 Models performance

In case of subtask 1.1, the acquired metrics for both assistant models demonstrate considerable promise. The model employed in this scenario was not pretrained, relying solely on its foundational capabilities as a Large Language Model (LLM). As anticipated, the model utilizing a knowledge base yielded superior results. We achieved a weighted F1-score of 0.52, accompanied by recall and precision values of 0.52 and 0.55, respectively.

For subtask 1.2, obtained metrics presented in Table 7 are much lower than metrics obtained in the SpanBERT case, where results on the SQuAD 1.1 were EM: 85.49, F1: 91.98. Given the nature of such models, metrics on our dataset should be

close to the baseline set by SpanBERT.

4.3.2 Limitations and future work

As for subtask 1.1, textual data makes determining expressed emotion challenging due to the absence of non-verbal cues. With abundant data and resources for fine-tuning, models can predict emotions more efficiently. Despite precise instructions, models may occasionally "hallucinate" and provide unsuitable answers, interpreting emotions differently from the defined set of six instructions.

Much more attention should be paid to preprocessing and analyzing the train, val, and test dataset in subtask 1.2 to provide more meaningful and balanced questions and answers in a given context. The provided sizes of all datasets could be much higher to utilize fine-tuning training fully. The training parameters should also be applied more carefully, and hyperparameter tuning should also be used.

5 Evaluation and Conclusions

For the evaluation phase, we have used the evaluation data provided by the Organizers. The data was emotion-classified using GPT-3, and the data was suited for span extraction as in Section 3.2. We have also tried to use BiLSTM in this case, however, its capabilities were very limited when processing data with unknown words and short sentences (the probability of each occurrence of emotion was nearly identical). The results from SpanBERT were answers to questions built upon classified emotions. Obtained answers were added to the original evaluation dataset's utterances (called *main utterances* in the following text) based on the created by Author unique ID. Answers also contained spans of text that could occur in different utterances, so the utterances that did not belong to the main utterance were placed in different lists (meaning multiple cause spans) in each matched main utterance. Based on the main utterance answer and additional answers, emotion-cause pairs were created in a manner that the "emotion-cause_pairs" list contained emotion utterance along with its emotion category and a cause utterance ID followed by position indexes of predicted cause span within the utterance. The position index starts from 0, and the ending index is the index of the last token plus 1 excluding the punctuation token at the beginning and end. The evaluation phase ended for 1 entry uploaded on the CodaLab (Pavao et al., 2023) submission as follows: **w-avg. Strict F1**: 0.0449, **w-avg. Proportional F1**: 0.0723, **Strict F1**: 0.0462, **Proportional F1**: 0.0717, resulting in 9th place out of 29 teams. The results showed that each of the presented subtasks, namely emotion classification and emotion-cause pair extraction, could perform better in terms of classifying emotions and extracting spans. The changes could improve the overall score by employing GPT-4 architecture and experimenting with span extraction model architecture as well.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Shunjie Chen, Xiaochuan Shi, Jingye Li, Shengqiong Wu, Hao Fei, Fei Li, and Donghong Ji. 2022. Joint alignment of multi-task feature and label spaces for emotion cause pair extraction.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020a. ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3161–3170, Online. Association for Computational Linguistics.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020b. End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3574–3583, Online. Association for Computational Linguistics.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

A BiLSTM for emotion recognition in conversations

For the emotion classification task, the BiLSTM model was tested (Schuster and Paliwal, 1997).

A.1 Preprocessing

Preprocessing for this task was separated into a few different steps:

1. Duplicates elimination, as we have discovered that sometimes data might have duplicates within;
2. Special signs and stopword removal - in this case, punctuation and digits were removed from the text, then data was converted to lowercase, split into a list of words, and cleaned from English stopwords obtained from the NLTK library;
3. Text tokenization, indexing, and text to-sequence conversion - vectorization was done on the text by turning text into a vector based on TF-IDF and by fitting it to the processed text;
4. Sequence padding, to make sure that all of the input sequences are of the same length;

A.2 Model

The training dataset, presented by the SemEval competition's creators, contained information about a conversation between some group of people and emotion-cause pairs extracted from that conversation. The conversation consisted of multiple utterances, each with defined text, speaker, and emotion expressed by the speaker and their id.

For this task, such dataset was partitioned with a ratio of 0.6-0.15-0.25 to create, train, validate, and test datasets. Such a dataset was transformed into a set of objects, where each represents a single utterance along with information about the context (concatenated utterances within the conversation) and expressed emotion.

Two configurations were checked to establish which parameters would give the best result. All of the configurations used categorical cross-entropy (Lin et al., 2017) as a loss function, Adam (Kingma and Ba, 2014) as an optimization algorithm, and f1_score, accuracy, and recall were noted during all of the training stages. The model in each configuration had seven outputs, each for every primary emotion (Ekman, 1992), including neutral.

Layer (type)	Output Shape
utterance_input (InputLayer)	[(None, 250)]
context_input (InputLayer)	[(None, 250)]
embedding_12 (Embedding)	(None, 250, 250)
embedding_13 (Embedding)	(None, 250, 250)
concatenate_6 (Concatenate)	(None, 250, 500)
bidirectional_1 (Bidirectional)	(None, 250, 150)
attention_1 (Attention)	(None, 250, 150)
concatenate_7 (Concatenate)	(None, 250, 300)
global_max_pooling1d_1 (GlobalMaxPooling1D)	(None, 300)
dense_5 (Dense)	(None, 64)
dropout_2 (Dropout)	(None, 64)
dense_6 (Dense)	(None, 32)
dropout_3 (Dropout)	(None, 32)
dense_7 (Dense)	(None, 7)

Table 8: Second BiLSTM Model with Attention layer configuration

The first configuration (similar, but less complex than presented in Table 8) was a BiLSTM with two hidden layers and ReLU set an activation function, with an attention layer (for utterance data and without context) set with softmax as an activation function.

The second configuration shown in Table 8 was a BiLSTM with two hidden layers and ReLU set an activation function, with an attention layer for contextual data set with softmax as an activation function and an additional layer for 1D convolution operation.

A.3 Evaluation

We have trained both models using train and validation datasets and then tested them using the corresponding set.

A.3.1 Metrics

During the testing phase, loss function and accuracy were calculated for training data, and for the validation data were also calculated recall and f1 score. A confusion matrix was created for the test data.

A.3.2 Training and testing

Both models show the same tendencies for the training data, with the loss function decreasing with each epoch and accuracy getting better (Figure 7).

However, looking at the accuracy, f1 score, and recall, their values are pretty similar for the data in the same batch (Model 1 or Model 2); results for Model 2 are significantly better (Figure 8).

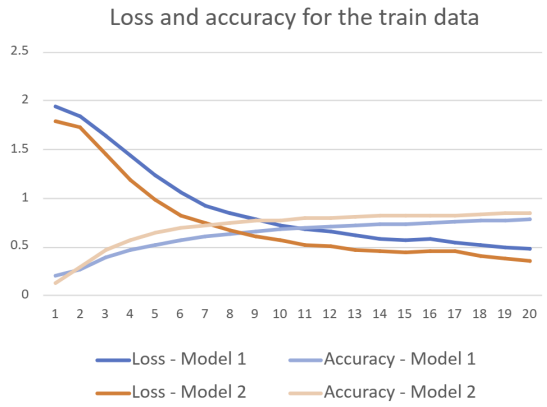


Figure 7: Metrics for test dataset in relation to the number of the epoch

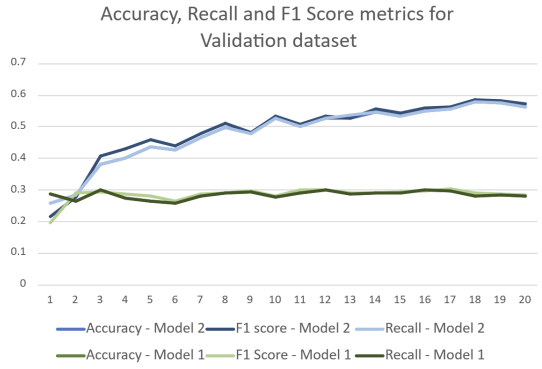


Figure 8: Metrics for validation dataset in relation to the number of the epoch

IUST-NLPLAB at SemEval-2024 Task 9: BRAINTEASER By MPNet (Sentence Puzzle)

Mohammad Hossein Abbaspour, Erfan Moosavi Monazzah, Sauleh Eetemadi
Iran University of Science and Technology
m_abbaspoor80, moosavi_m@comp.iust.ac.ir
sauleh@iust.ac.ir

Abstract

This study addresses a task encompassing two distinct subtasks: Sentence-puzzle and Word-puzzle. Our primary focus lies within the Sentence-puzzle subtask, which involves discerning the correct answer from a set of three options for a given riddle constructed from sentence fragments. We propose four distinct methodologies tailored to address this subtask effectively. Firstly, we introduce a zero-shot approach leveraging the capabilities of the GPT-3.5 model. Additionally, we present three fine-tuning methodologies utilizing MPNet as the underlying architecture, each employing a different loss function. We conduct comprehensive evaluations of these methodologies on the designated task dataset and meticulously document the obtained results. Furthermore, we conduct an in-depth analysis to ascertain the respective strengths and weaknesses of each method. Through this analysis, we aim to provide valuable insights into the challenges inherent to this task domain.

1 Introduction

The remarkable efficacy of language models in navigating complex reasoning tasks, particularly in the realm of vertical thinking, has prompted their exploration in lateral thinking problem domains (Waks, 1997). One such domain, exemplified by the BRAINTEASER task (Jiang et al., 2024), entails a multiple-choice Question Answering framework comprising two distinct subtasks: Sentence-puzzle and Word-puzzle. This paper directs its focus toward the Sentence-puzzle subtask, which hinges on unraveling the intricate nuances of common sense embedded within sentence fragments (Jiang et al., 2023).

Initially, we adopted a zero-shot approach, followed by experimentation with three distinct fine-tuning methodologies tailored for Language Model (LLM) architectures as the backbone. Additionally, we have made our code openly accessible on GitHub[1] to facilitate reproducibility and further research endeavors.

A primary challenge we encountered pertained to the constraint imposed by the dataset size, posing impediments to both fine-tuning procedures and model training from scratch. To mitigate this challenge, we employed various strategies, including the utilization of k-fold cross-validation techniques, to enhance the robustness and generalizability of our approach.

2 Background

In the implementation of the zero-shot method, we employed ChatGPT-3.5, utilizing a consistent prompt template throughout. Conversely, for the fine-tuning process, we adopted a pre-trained sentence embedding technique to map input sentences into meaningful numerical vectors, facilitating subsequent decision-making regarding the provided questions and options.

Furthermore, in our fine-tuning methodologies, we integrated two distinct types of loss functions: Binary Cross-Entropy loss and Triplet loss. The Binary Cross-Entropy loss function operates on the premise of determining whether two sentences coherently match or not. Conversely, the Triplet loss function aims to optimize the proximity between the question and the correct answer while concurrently ensuring a clear distinction between the question and unrelated options.

3 Dataset

The task dataset comprises 507 samples designated for training purposes, with an additional 120 samples allocated for the test set. Notably, the evaluation set encompasses a subset of the training samples, necessitated by data scarcity. Consequently, for two out of the three fine-tuning methods, no

[1] https://github.com/MohammadHAbbaspour/SemEval-2024_task9_BRAINTEASER

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1106–1109

data from the training set were utilized for evaluation. Conversely, the third method, employing the k-fold technique, leveraged the training samples for both the training and evaluation phases.

4 System overview

4.1 Zero Shot

For the zero-shot methodology, we leveraged the ***GPT-3.5-turbo*** model, utilizing a temperature parameter of 0.0. To elicit responses from the model, we employed a consistent prompt template outlined in Table 1. Within this template, we systematically substituted the question and available options with the corresponding tokens. Additionally, we extracted explanations from the model to facilitate a deeper analysis of its reasoning processes.

4.2 Binary Classification

In this approach, we utilized the ***all-mpnet-base-v2*** model (Song et al., 2020; Jayanthi et al., 2021) as the backbone, which was subsequently frozen. Following this, we introduced a trainable layer for inference purposes. The core principle underlying this method involves the transformation of each sample within the dataset, comprising a question and three options (excluding the 'None of the above' option), into three distinct pairs. Each pair encompasses the question alongside one of its options, with a corresponding label indicating whether the option constitutes the correct answer. Consequently, the original training dataset, comprising 507 samples, was expanded to form a new dataset comprising 1521 samples.

Moreover, during the process of feeding sentences into the model, we initially present the question and option to the backbone model. Subsequently, we concatenate the resulting vectors and forward them to the inference layer. For the final decision-making step, we apply a sigmoid function to the output of the inference layer, enabling us to ascertain the consistency between the two sentences by employing a threshold of 0.5.

During the inference stage, we determine the option with the highest score among the three available options.

4.3 Triplet loss

In this approach, our base model and backbone remain consistent with the previous section. However, the data preparation process differs. In the original dataset, each sample consists of a single question alongside three options, one of which is designated as the correct answer. Consequently, for each sample in the original dataset, we generate two samples in the new dataset. As a result, the new dataset comprises 1014 samples, with each sample comprising a question as the anchor, the correct answer as the positive, and a distractor as the negative.

As previously elucidated, the fundamental concept is to minimize the distance between the question and the correct answer while maximizing the distance between the question and unrelated options. To achieve this objective, we integrate a pre-trained sentence embedding model within the inference component.

Within our implementation, the inference component consists of two subparts: one dedicated to the anchor and the other to the positive and negative instances. The anchor component essentially functions as an identity layer, as it cannot glean meaningful insights from a single question. Conversely, the other component aims to discern the disparities between the positive and negative instances by leveraging information from the question. Hence, we concatenate the output of the sentence embedding model for the question and the correct answer to form the positive instance, and likewise for the question and the distractor to constitute the negative instance within the triplet loss framework (see Algorithm 1).

Algorithm 1 Algorithm of the triplet loss

procedure FORWARD($qemb, ansemb, disemb$)
    anchor = $qemb$
    positive = concatenate($qemb$, $ansemb$)
    negative = concatenate($qemb$, $disemb$)

    anchor = anchor_inference(anchor)
    positive = option_inference(positive)
    negative = option_inference(negative)
return anchor, positive, negative

4.4 Triplet loss (K-Fold)

As previously highlighted, the limited availability of data poses a significant challenge in this task. To address this issue, we adopt the K-Fold technique, a commonly employed strategy for mitigating data scarcity. The key components of this approach remain consistent with the previous sections, including the backbone model and the underlying algorithm. However, the distinguishing factor lies

Prompt

```
Which option is the answer to this riddle, explain in a step-by-step manner:
<RIDDLE>
1) <OPTION1>
2) <OPTION2>
3) <OPTION3>
4) None of the above.
 Please place your answer in a JSON format:
{
  "option_number": <JUST_THE_NUMBER_OF_THE_CORRECT_OPTION>,
  "explanation": <EXPLANATION_WHY_IT_IS_CORRECT>
}
```

Table 1: Constant template for prompt used in zero-shot

in the training process, wherein multiple models are trained, and the most performant one is selected as the final iteration based on its performance on the evaluation data.

Initially, we partition the training dataset into **k** folds, each serving as the basis for training a distinct model utilizing the Triplet loss. Subsequently, a subset of validation data is extracted from each fold, and the model is trained on the remaining data. Evaluation of each model is then conducted on the evaluation data corresponding to its respective fold. Upon completion of the training process, **k** models are obtained. To select the final model, we employ a sorting criterion based on the following key metric:

$$key = \frac{val\_acc}{train\_loss}$$

This metric encapsulates the trade-off between validation accuracy and training loss. The selection process involves sorting the models based on this key metric and choosing the middle model. This decision is predicated on the objective of maximizing validation accuracy while minimizing training loss. However, it is important to note that models with the highest values of **key** may exhibit signs of overfitting and possess reduced generalization capabilities. Hence, opting for the middle model mitigates the risk of overfitting and ensures enhanced generalization.

5 Experimental setup

5.1 Customized triplet loss

In methodologies utilizing the triplet loss paradigm, the loss function undergoes customization. While the original triplet loss hinges on the calculation of the **Euclidean distance** as a measure of difference, our approach diverges by customizing this metric to **cosine similarity** (see Algorithm 2).

Algorithm 2 Triplet loss, customized by cosine similarity

procedure LOSS($anchor, positive, negative$)
positive_sim = cosine_similarity($anchor$, $positive$)
negative_sim = cosine_similarity($anchor$, $negative$)
loss = $negative\_sim$ - $positive\_sim$ + $margin$
return loss

The maximum value of cosine similarity between two vectors is 1 which means two vectors are the same, and the minimum value between them is -1 which means they are different. So:

$$-2 \leq positive\_sim - negative\_sim \leq 2$$

We add the **margin=2** value to the loss for shifting it in positive numbers:

$$0 \leq positive\_sim - negative\_sim + margin \leq 4$$

Hence, if two vectors are the same it means the loss is equal to 0 and if two vectors are opposite it means the loss has its max value.

5.2 Hyperparameters

For the training of the discussed models, we scrutinized the hyperparameters outlined in Table 2.

6 Results

We conducted evaluations of the aforementioned methods on the test set, and the results are presented in Table 3. It is evident that the zero-shot method exhibits the best performance on the

	epochs	learning rate	batch size	validation size	k
Binary classification	10	0.001	4	-	-
Triplet loss	10	0.001	16	-	-
K-Fold	10	0.001	16	20	3

Table 2: Hyperparameters of models while training

	S_ori	S_sem	S_con	S_ori_sem	S_ori_sem_con	S_overall
Zero-shot	**0.7**	0.575	0.575	0.55	0.35	0.61
Binary classification	0.625	0.625	0.525	0.625	0.475	0.5916
Triplet loss(modified system)	0.65	**0.65**	**0.625**	**0.65**	**0.525**	**0.641**
K-Fold	0.6	0.6	**0.625**	0.6	0.5	0.608

Table 3: Comparison of our results

Original sentences. However, its efficacy diminishes notably when applied to other categories such as **Semantic**, showcasing a significant disparity compared to its performance on the **Original sentences**. Conversely, all of our fine-tuning methods demonstrate comparable performance across all categories.

Among our fine-tuning methodologies, the **Triplet loss** approach stands out with the most impressive performance, achieving the highest **Overall** score among all methods.

The uniformity in scores observed with the **Triplet loss** method suggests that it does not exhibit bias towards specific words or segments of the sentence; rather, it considers the entire sentence holistically. This is in contrast to the zero-shot method, where significant discrepancies exist among its scores. However, it's worth noting that the performance of our fine-tuning models could potentially improve with a larger volume of data.

7 Conclusion

In this paper, we have presented four distinct methodologies for the BRAINTEASER task, a novel challenge involving common sense reasoning and sentence puzzle solving. We have evaluated our methods on the task dataset and compared their performance across different categories. Our results show that the zero-shot approach, based on GPT-3.5-turbo, achieves the highest score on the original sentences, but fails to generalize well to other categories. On the other hand, our fine-tuning methods, based on MPNet and various loss functions, demonstrate more consistent and robust performance across all categories, with the triplet loss approach achieving the best overall score. We have also employed the K-Fold technique to mitigate the data scarcity issue and enhance the generalization capability of our models. Through our analysis, we have provided valuable insights into the strengths and weaknesses of each method, as well as the challenges inherent to this task domain. We hope that our work will inspire further research on this novel and intriguing problem of common sense reasoning and sentence puzzle solving.

References

Sai Muralidhar Jayanthi, Varsha Embar, and Karthik Raghunathan. 2021. Evaluating pretrained transformer models for entity linking in task-oriented dialog.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6(4):245–255.

iimasNLP at SemEval-2024 Task 8: Unveiling structure-aware language models for automatic generated text identification

Andric Valdez,
Posgrado en Ciencia e Ingeniería de la Computación
Helena Gomez-Adorno, Fernando Márquez, Jorge Pantaleón,
Instituto de Investigaciones en Matemáticas y en Sistemas
Gemma Bel-Enguix
Instituto de Ingeniería
Universidad Nacional Autónoma de México, Ciudad de México
andric_valdez@comunidad.unam.mx

Abstract

Large language models (LLMs) are artificial intelligence systems that can generate text, translate languages, and answer questions in a human-like way. While these advances are impressive, there is concern that LLMs could also be used to generate fake or misleading content. In this work, as a part of our participation in SemEval-2024 Task-8, we investigate the ability of LLMs to identify whether a given text was written by a human or by a specific AI. We believe that human and machine writing style patterns are different from each other, so integrating features at different language levels can help in this classification task. For this reason, we evaluate several LLMs that aim to extract valuable multilevel information (such as lexical, semantic, and syntactic) from the text in their training processing. Our best scores on SubtaskA (monolingual) and SubtaskB were 71.5% and 38.2% in accuracy, respectively (both using the ConvBERT LLM); for both subtasks, the baseline (RoBERTa) achieved an accuracy of 74%.

1 Introduction

Large language models (LLMs) have become widely available and easily accessible, leading to an increase in machine-generated content across diverse platforms including question-and-answer forums, social media platforms, educational resources, and academic settings.

Recent advancements in LLM technology, exemplified by models like ChatGPT and GPT-4, produce coherent responses to a vast majority of user inquiries, making them increasingly appealing for replacing human labor in various applications. However, this accessibility has raised concerns about potential misuse, such as generating fake news, financial services industry, legal domain, and disruptions in educational settings. Given the challenge humans face in distinguishing between machine-generated and human-written text, there is a pressing need to develop automated systems capable of identifying machine-generated content to mitigate the risks associated with its misuse.

Motivated by these challenges, SemEval-2024 Task-8 (Wang et al., 2024) offers three subtasks over two paradigms of text generation: (1) full text when a considered text is entirely written by a human or generated by a machine; and (2) mixed text when a machine-generated text is refined by a human or a human-written text paraphrased by a machine.

These three subtasks are composed in the following way: Subtask A is a binary classification task that focuses on identity if a given text was written by a human or a machine; it is split into monolingual (English) and multilingual (Arabic, Russian, Chinese, etc). Subtask B is a multi-class classification task that aims to identify which specific LLM generates a given text among six different known options: Human-made, ChatGPT, Cohere, DaVinci, Bloomz, and Dolly. Finally, Subtask C, given a mixed text, where the first part is human-written and the second part is machine-generated, determines the boundary, where the change occurs.

We tackled two of these three subtasks: Subtask A (monolingual) and Subtask B. We applied fine-tuning of four LLMs (described in the following section) that included structural information in their pre-training. These models have proven their efficiency in multiple Natural Language Understanding (NLU) tasks, such as question-answer entailment, paraphrasing, and textual similarity. We aim to test the efficiency in machine-text detection by comparing the results of given baselines for each subtask (A and B) with our fine-tuning LLMs with different approaches for the implementation of structural information.

Our scores show a modest performance related to the final ranking (especially in Subtask B), but, based on the analysis of the results We observe that all of these LLMs used in this research, struggle

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1110–1114

to classify human text, meanwhile, they achieve a good performance classifying machine text.

This paper is structured as follows: Section 2 summarizes related works on machine text generation. Section 3 describes the dataset used for the task. Section 4 presents the system overview and the experimental setup. Section 5 and 6 shows the results and conclusions, respectively.

2 Related Work

In recent years, many interesting shared tasks that related to the automatic detection of AI-generated text. Besides the SemEval task8, one of the most popular and challenging tasks called Autextification: Automated Text Identification (Sarvazyan et al., 2023), aims to address the detection of content created by text generation models in English and Spanish.

To mention a few interesting research works related to Autextification-2023, the system titled "I've Seen Things You Machines Wouldn't Believe: Measuring Content Predictability to Identify Automatically Generated Text" (Przybyła et al., 2023) achieves the best performance among the submissions in subtask 1 (differentiating between human- and machine-generated text), both for English and Spanish. Their model focuses on assessing the "predictability" of given text by multiple LLMs, leveraging features related to grammatical accuracy, word frequency, and linguistic patterns, along with a fine-tuned LLM representation. Another remarkable work titled "Generative AI Text Classification using Ensemble LLM Approaches" (Abburi et al., 2023), proposes an ensemble neural model that leverages probabilities generated by different pre-trained LLMs as features for a Traditional Machine Learning (TML) classifier (their model ranked in first place in subtask 2 for English and Spanish).

On the other hand, pre-training LLMs with structural information enrich the learning process with contextual and syntactic cues. These cues encompass sentence structure, paragraph organization, grammatical rules, and broader linguistic patterns. Fine-tuning LLMs with such structural knowledge enhances their ability to both comprehend and generate text that adheres to human-like writing styles and conventions.

This approach has been explored in multiple ways; so now we briefly describe the approach taken by the models we used: ERNIE model (Sun et al., 2021) implements an implicit knowledge of syntactic information through multiple levels of masking (token, phrase, and entity level). SpanBERT model (Joshi et al., 2020) masks random spans of contiguous tokens and trains to predict every token for each span instead of just masking and predicting each token. ConvBERT model (Jiang et al., 2020) substitutes attention blocks for span-based dynamic convolutions capable of storing structural information in the generated kernels. Finally, XLNet (Yang et al., 2019), this LLM does not corrupt the text with masking but rather utilizes all the multiple permutations of tokens in a given sentence during the training process.

3 Dataset

The data provided for SemEval Task 8 is an extension of the M4 dataset (Wang et al., 2023). This is a large-scale benchmark, which is a multi-generator, multi-domain, and multi-lingual corpus for machine-generated text detection. This extensive M4 corpus encompasses texts from various domains, including news articles, programming code, and fictional narratives. Additionally, the M4 corpus incorporates texts in numerous languages, such as English, Spanish, and Chinese. This diversity in both domain and language coverage contributes to the effectiveness of M4 in effectively identifying machine-generated text (see figure 1).

For machine generation, it prompts the following multilingual LLMs: GPT-4, ChatGPT, GPT3.5 (tex-davinci-003), Cohere, and Dolly-v2. The models are asked to write articles given a title (Wikipedia), abstracts given a paper title (arXiv), peer reviews based on the title and the abstract of a paper (PeerRead), news briefs based on a title (news), also to summarize Wikipedia articles (Arabic), and to answer questions (Reddit).

Source/ Domain	Language	Total Human	Parallel Data Human	Davinci003	ChatGPT	Cohere	Dolly-v2	BLOOMz	Total
Wikipedia	English	6,458,670	3,000	3,000	2,995	2,336	2,702	3,000	17,033
Reddit ELI5	English	558,669	3,000	3,000	3,000	3,000	3,000	3,000	18,000
WikiHow	English	31,102	3,000	3,000	3,000	3,000	3,000	3,000	18,000
PeerRead	English	5,798	5,798	2,344	2,344	2,344	2,344	2,344	17,518
arXiv abstract	English	2,219,423	3,000	3,000	3,000	3,000	3,000	3,000	18,000
Baike/Web QA	Chinese	113,313	3,000	3,000	3,000	–	–	–	9,000
RuATD	Russian	75,291	3,000	3,000	3,000	–	–	–	9,000
Urdu-news	Urdu	107,881	3,000	–	3,000	–	–	–	9,000
id_newspapers_2018	Indonesian	499,164	3,000	–	3,000	–	–	–	6,000
Arabic-Wikipedia	Arabic	1,209,042	3,000	–	3,000	–	–	–	6,000
True & Fake News	Bulgarian	94,000	3,000	3,000	3,000	–	–	–	9,000
Total			35,798	23,344	32,339	13,680	14,046	14,344	133,551

Figure 1: Statistics about our M4 dataset, which includes non-parallel human data and parallel human and machine-generated texts.

4 System overview

Our system evaluates different LLMs that integrate features at different language levels (such as lexical, semantic, and syntactic) with the idea of extracting human and machine writing style patterns and being able to distinguish text from each other. For this reason, We applied a fine-tuning process using the four LLMs mentioned before: ERNIE[1], SpanBERT[2], ConvBERT[3], and XLNet[4] (using the Hugging Face library) for Subtask A Monolingual and Subtask B.

Starting with the data partition process, We used the same partition proposed by the organizers in the baseline code for both tasks. The training dataset was split into the train (80%) and validation (20%) for the fine-tuning process and the development dataset was used to measure the accuracy of each model with unknown data. Finally, the test dataset was only used to rank the models and verify the results.

Afterwards, in the fine-tuned process we tried with different hyperparameters on batch size (16, 32), learning rates (2e-5, 5e-5), random seed (0, 42), epochs (3, 5), and a weight decay of 0.01. Along with these params configurations, we used the Trainer, AutoModel, and AutoTokenizer classes from the Transformers. Each sequence was padded and truncated at 512 after tokenization due to the constraints of some of the models we used (most of them had a limit in the allowed length of the input sequence). These hyperparameters were chosen based on empirical experiments and hyperparameter tuning to achieve the best performance on our validation dataset. For the evaluation we computed macro-F1, micro-F1, and accuracy scores; being the last ones used by those organized to evaluate the final ranking.

Finally, in the test process, the output predictions for the model (logits) serve as an input for a Softmax function and then apply an argmax function in order to get the final prediction class.

5 Results

After the fine-tuning process using the training dataset, we measured the performance of each LLMs on the test set (development set provided) for Subtask A (Monolingual) and Subtask B (using the respective training and test data provided).

Table 1 shows the evaluation results for Subtask A (Monolingual) using the macro-F1, micro-F1, and accuracy measures (obtained from the score scripts provided for the organizers). For the Validation set, ERNIE's model outperforms the other LLMs across all metrics achieving 79.4 % in accuracy, but, ConvBERT and SpanBERT closely follow with 77.1% and 78.8% respectively.

For subtask A (Monolingual), We submitted our two best prediction results to the Codabench platform: ERNIE and ConvBERT LLMs. Table 3 shows the final ranking for this Subtask, we ranked place 87 out of 137 with an accuracy score of 71.5% (obtained by the ConvBERT LLM). The best team (safeai) obtained an accuracy score of 96.8% and the baseline (RoBERTa LLM) achieved 88.4%. Table 1 also shows the results evaluating these models in the test set (post-submission, using gold labels released by organizers); in this case, our best model was the ConvBERT with 77.6% in accuracy.

On the other hand, Table 2 shows the performance metrics obtained for SubTask B. In the Validation set, the SpanBERT model outperforms the other LLMs across all metrics achieving 66.8% of accuracy, 66.8% of micro-F1, and 63.4% of macro-F1 score. However, the ERNIE model closely follows with 65.4% accuracy; Then, we obtained the final predictions from the validation dataset using these fine-tuned trained models and uploaded to Codabench platform one submission based on the ConvBERT LLM results. Table 2 shows the final ranking for Subtask B, where we obtained place 67 out of 77 with an accuracy score of 38.2% (obtained by the ConvBERT LLM). In this case, the best team (tmarchitan) achieved an accuracy score of 86.9% and a baseline (RoBERTa LLM) of 74.6%. Finally, as in Subtask A, We re-evaluated these models on the test set released (post-submission), and, our best model was the XLNET with 65.2% in accuracy (second part in table 2).

On the other hand, figure 2 and figure 3 show the Confusion Matrix (CM) results for Subatsk A Monolingual and Subtask B, respectively. The CM for Subtask A across all models, presents a large confusion in classifying human text (True Positive vs False Positive) compared to the performance achieved for the machine-generated text (True Negative vs False Negative). For human text classification, the ConvBERT LLM was the best model

[1]https://huggingface.co/nghuyong/ernie-2.0-base-en
[2]https://huggingface.co/SpanBERT/spanbert-base-cased
[3]https://huggingface.co/YituTech/conv-bert-base
[4]https://huggingface.co/xlnet/xlnet-base-cased

SubTask A (Monolingual)					
Dataset	**Measure**	**Large Language Model**			
		ERNIE	**SpanBERT**	**ConvBERT**	**XLNET**
Validation Set	***macro-F1***	**0.789**	0.783	0.762	0.720
	micro-F1	**0.794**	0.788	0.771	0.733
	accuracy	**0.794**	0.788	0.771	0.733
Test Set*	***macro-F1***	0.701	0.760	**0.770**	0.758
	micro-F1	0.720	0.772	**0.776**	0.767
	accuracy	0.720	0.772	**0.776**	0.767

Table 1: Results obtained for each LLM on the Validation and Test set for Subtask A (monolingual).
* These results were obtained after the competition was finalized.

SubTask B					
Dataset	**Measure**	**Large Language Model**			
		ERNIE	**SpanBERT**	**ConvBERT**	**XLNET**
Validation Set	***macro-F1***	0.620	**0.634**	0.615	0.601
	micro-F1	0.654	**0.668**	0.640	0.634
	accuracy	0.654	**0.668**	0.640	0.634
Test Set*	***macro-F1***	0.578	0.518	0.603	**0.590**
	micro-F1	0.626	0.563	0.634	**0.652**
	accuracy	0.626	0.563	0.634	**0.652**

Table 2: Results obtained for each LLM on the Validation and Test set for Subtask B.
* These results were obtained after the competition was finalized.

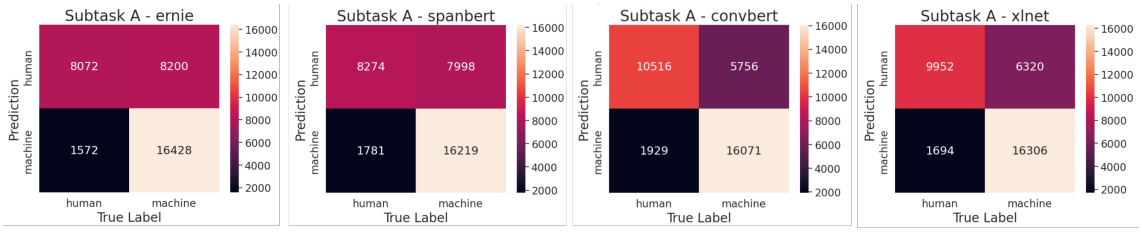


Figure 2: Subtask A Monolingual. Confusion Matrix results for each LLM applied on the test set (post-submission).

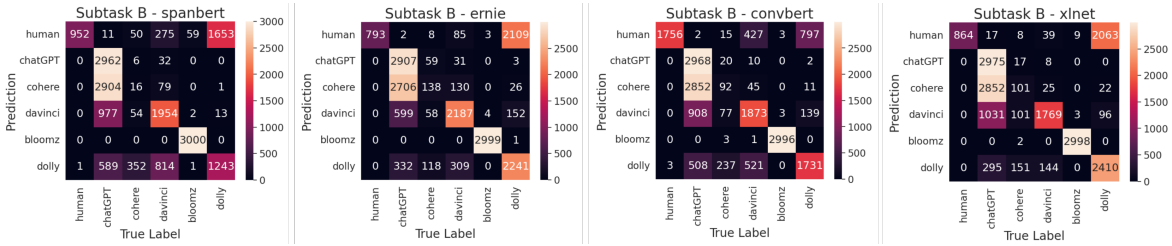


Figure 3: Subtask B. Confusion Matrix results for each LLM applied on the test set (post-submission).

getting 65% correct and 35% fail, meanwhile, the ERNIE LLM obtained a poor performance of 50% correct and 50% fail. Related to machine-generated text classification, all models performed similarly, obtaining less confusion: around 90% correct and 10% fail. Furthermore, CM for Subtask B in general struggles to classify human texts and presents a large confusion with dolly machine model across all LLM; the best performance was classifying chatGPT and bloomz text, getting around 98% correct in both, meanwhile, cohere and dolly machine obtained a poor classification performance.

Finally, We would like to mention that, due to some technical issues in our servers, We did not submit the models with the best model scores in the validation stages for both subtasks. For this reason,

Position	Team	Accurary
1	safeai	0.968
2	comp5	0.960
19	baseline	0.884
87	**iimasNLP (andric)**	**0.715**
137	saibewaraditya	0.231

Table 3: Final ranking per team in Subtask A (monolingual)

Position	Team	Accurary
1	tmarchitan	0.869
2	farawayxxc	0.843
24	baseline	0.746
67	**iimasNLP (andric)**	**0.382**
77	saibewaraditya	0.153

Table 4: Final ranking per team in Subtask B

We reported different scores in the final submission compared to our scores in the Test evaluation (post-submission, with gold labels).

6 Conclusion

We applied a fine-tuning process using four LLMs: ERNIE, SpanBERT, ConvBERT, and XLNet. In general, this LLM aims to extract lexical, semantic, and syntactic information from the text. We obtained comparable results with the baselines reported (initially), but, below compared to those in the first positions.

For future work, it could be interesting to prove more LLMs that focus on multilevel language and stylistic features; also apply a more robust finetuning process to evaluate more hyperparameters; and finally try a different approach based on text graph called Graph Neural Networks.

Acknowledgements

This paper has been supported by PAPIIT-UNAM projects IN104424, TA101722, and CONAHCYT CF-2023-G-64. The authors thank Ricardo Villareal and Rita Rodriguez for the technical support with computational resources and Roman Osorio for the student administration support. This work has the support of the CONAHCyT graduate scholarship program.

References

Harika Abburi, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. Generative ai text classification using ensemble llm approaches. *arXiv preprint arXiv:2309.07755*.

Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. 2020. Convbert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Piotr Przybyła, Nicolau Duran-Silva, and Santiago Egea-Gómez. 2023. I've seen things you machines wouldn't believe: Measuring content predictability to identify automatically-generated text. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023). CEUR Workshop Proceedings, CEUR-WS, Jaén, Spain*.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains. In *Procesamiento del Lenguaje Natural*, Jaén, Spain.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

INGEOTEC at SemEval-2024 Task 10: Bag of Words Classifiers

Mario Graff[‡] and **Eric S. Tellez**[‡] and **Mireya Paredes**[‡]
and **Daniela Moctezuma**[†] and **José Ortiz-Bejar**[⋆]
[†] CentroGEO, Aguascalientes, México
[‡] CONACyT - INFOTEC, Aguascalientes, México
[⋆] Universidad Michoacana de San Nicolás de Hidalgo, Morelia, México
{mario.graff,eric.tellez,mireya.paredes}@infotec.mx
dmoctezuma@centrogeo.edu.mx
jose.ortiz@umich.mx

Abstract

The Emotion Recognition in Conversation subtask aims to predict the emotions of the utterance of a conversation. In its most basic form, one can treat each utterance separately without considering that it is part of a conversation. Using this simplification, one can use any text classification algorithm to tackle this problem. This contribution follows this approach by solving the problem with different text classifiers based on Bag of Words. Nonetheless, the best approach takes advantage of the dynamics of the conversation; however, this algorithm is not statistically different than a Bag of Words with a Linear Support Vector Machine.

1 Introduction

Sentiment analysis has been very useful for emotion detection in digital text. For example, in the analysis of a customer review, a sentiment analysis system can find whether the review is positive or negative. Today, this way of finding out what the sentiment expressed in the digital text has been popular due to its potential to have a feeling over what people are writing about. Recent studies have been conducted towards sentiment analysis, not only in a one-party text (text written by one person/source) but also within a multi-party conversational text (Hazarika et al., 2018)(Majumder et al., 2018)(Poria et al., 2019), known as Emotion Recognition in Conversation (ERC). ERC refers to the emotion detection of each of the phrases/utterances within a dialogue. For instance, in a conversation between two people (speaker 1, SP1, and speaker 2, SP2) saying the following, SP1: "I had an awful day", SP2 replies "Oh no, what happened?". SP1 may have a "sad" emotion and SP2 may be also "sad". However, following the conversation SP1 answers "Somebody ate my sandwich!" and SP2 replies "I can make you a new one right now!)". This answer provokes a change in the emotion of SP1 to "joy". The aim of ERC is precisely the detection of speakers' emotional changes involved within a dialogue. ERC research has become popular due to the vast amount of conversation sources in social media such as opinion mining in chat history, social media threads, debates, and understanding consumer feedback in live conversations, among others (Majumder et al., 2018).

To date, several studies have investigated ERC using different approaches (Hazarika et al., 2018)(Majumder et al., 2018) as it is summarized by (Poria et al., 2019). (Hazarika et al., 2018) presented a framework for emotion detection in conversations using a recurrent neural network (RNN) based memory network with multi-hop attention modeling. (Majumder et al., 2018) is a method based on an RNN that maintains information of each party separately and this information is used for emotion classification. Most recent studies have been focused on more elaborated proposals about emotion detection based on the context and the common sense knowledge within the conversation (Tu et al., 2022). Recently, (Jiang et al., 2024) presented a self-supervised model to better understand the semantics within the text associated with the order of the utterances.

In this paper, we present a model given the *Task 10: Emotion Discovery and Reasoning its Flip in Conversation(subtask 1)* of the SemEval-2024 workshop (Kumar et al., 2024) which consists of applying *Emotion Recognition in Conversation (ERC)* to Hindi-English code-mixed conversations. We propose to solve the challenge as a classification problem, using a Bag of Words (BoW) for the representation. Despite the usage of BoW is not so common anymore due to the current usage of more sophisticated techniques as deep learning, our work is built on the previous work presented in (Graff

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1115–1120
June 20-21, 2024 ©2024 Association for Computational Linguistics

et al., 2023a,b), leading to a unique and customized BoW for solving this specific problem.

The rest of the paper is organized as follows: Section 2 presents the background; Section 3 introduces a description of the model; finally Section 4 shows a brief analysis of the results and Section 5 concludes the paper.

kumar2024semeval

2 Background

Emotion recognition has been a popular research field using Artificial Intelligence. According to (Saxena et al., 2020), several methods are applied for Emotion recognition including facial expression recognition, physiological signals recognition, speech signals variation, and text semantics, among others. Specifically, in this work, we focus on emotion recognition written in a digital text. Previously, emotion recognition in the text has been focused on the selection of emotional keywords (Seol et al., 2008) and the classification of their emotional state within a conversation. However, this keyword-based method presented some limitations as ambiguity and the lack of semantic and syntactic information. Emotion recognition in a conversation has shown to be a challenge due to the way emotions change over time. Other machine learning method have been applied as ICON (Hazarika et al., 2018) and DialogueRNN (Majumder et al., 2018), both using RNN. ICON (Hazarika et al., 2018) generates memories from the conversation to generate a good context for predicting emotions within a conversational video. DialogueRNN (Majumder et al., 2018) presented a model involving three aspects in a conversation: the speaker, the context of the previous phrases, and the emotion, according to them, taking into account these key aspects leads to a much better context representation. Commonsense knowledge is something difficult to pass over a machine, for that reason new approaches added external knowledge to help the machine to have a context (Speer et al., 2018)(Cambria et al., 2022) for solving new challenges as having an empathetic dialogue system (Ma et al., 2020).

3 System overview

The subtask Emotion Recognition in Conversation can be posed as a supervised learning problem. Without considering that emotions are part of a conversation, the problem can be seen as finding the mapping between an utterance and its associated emotion, i.e., it is a classification problem. In order to use the majority of classifiers, one needs to transform the utterance into a format amenable to the classifier selected. Generally, the representation acceptable for the majority of traditional classifiers is vectors.

Perhaps one of the most studied representations that transform a text into a vector is the Bag of Words (BoW); it is not so common anymore because it has been overcome by the use of deep learning techniques such as the attention mechanism (Vaswani et al., 2017). However, our participation is based solely on the use of BoW, following a similar approach used in previous competitions see (Graff et al., 2023a,b), and complementing it with an approach tailored for this specific subtask.

The realm of the BoW representation is that each token t of a text is associated with a vector $\mathbf{v_t} \in \mathbb{R}^d$. In this contribution, the i-th component of $\mathbf{v_t}$ corresponds to the token's Inverse-Document-Frequency (IDF) estimated in a collection of Hindi tweets (9.5 million), and the rest of the components of $\mathbf{v_t}$ are zero, i.e., $\forall_{j \neq i} \mathbf{v_t}_j = 0$. The set of all tokens is fixed, and these correspond to the vocabulary. The vocabulary is fixed to containing only $2^{17} = d$ elements, and this corresponds to the most frequent tokens found in the collection of tweets. Furthermore, given that only one component of each vector is different from zero, the set of all the vectors constituted a basis, and each text is represented in this vector space. Additionally, any token that is not found in the vocabulary is discarded from the representation.

Using this notation, a text x is represented by the sequence of its tokens, i.e., $(t_1, t_2, \ldots)$; the sequence can have repeated tokens, e.g., $t_j = t_k$. Then each token is associated with its respective vector $\mathbf{v}$ (keeping the repetitions), i.e., $(\mathbf{v_{t_1}}, \mathbf{v_{t_2}}, \ldots)$. Finally, the text x is represented as:

$$\mathbf{x} = \frac{\sum_t \mathbf{v_t}}{\|\sum_t \mathbf{v_t}\|}, \tag{1}$$

where the sum goes for all the elements of the sequence, $\mathbf{x} \in \mathbb{R}^d$, and $\|\mathbf{w}\|$ is the Euclidean norm of vector $\mathbf{w}$. The term frequency is implicitly computed in the sum because the process allows token repetitions.

The second representation is inspired by a self-supervised technique, particularly the procedure of masking tokens in a text and then developing an algorithm to predict the masked tokens.

The idea is pursued by creating M binary classification problems where the task is to predict the presence of a particular token; in this case, the tokens are words (defined as a string surrounded by spaces or punctuation symbols) or emojis. The words are selected based on their frequency; the most frequent words are not considered, and the words considered started when the plot of rank vs. frequency settles, i.e., it is when the flat part starts. On the other hand, all the emojis are considered. However, only the words and emojis where there are more than 1024 positive examples in the collection are kept. In total, there are 176 tweets and 2048 words.

There are 2,224 binary classification problems; each is solved using a BoW representation where the classifier is a Linear Support Vector Machine. Consequently, there are M binary text classifiers, i.e., $(c_1, c_2, \ldots, c_M)$. The utterance is represented using the decision values of the M binary classifiers; that is, the text lives in $\mathbb{R}^M$. As can be seen, each component is associated with either a word or emoji, and its value indicates the likelihood of its presence in the text. We refer to this representation as Dense. Finally, the classifier used with the dense representation is again a Linear Support Vector Machine.

After the competition ended, we decided to include in the comparison a procedure to combine (using Stacking (Graff et al., 2020)) the BoW and Dense representation, namely StackBoW. The idea is to make a convex combination of the class probabilities predicted by these two classifiers. The approach is to use the training set with k-fold cross-validation to estimate the decision function of these two classifiers on the training set. These decision functions are then transformed with softmax to obtain probabilities, and then an optimizer is used to estimate the convex combination. There are 8 emotions, so the optimizer needs to find 8 coefficients, each corresponding to a class. For example, let $\mathbf{p}_b \in \mathbb{R}^8$ be the probability given by the BoW classifier, $\mathbf{p}_d \in \mathbb{R}^8$ corresponds to the dense classifier, and $\beta \in \mathbb{R}^8$ are the estimated coefficients, then the prediction of the StackBoW is $\beta \odot \mathbf{p}_b + (1-\beta) \odot \mathbf{b}_d$ where $\odot$ is the pointwise product.

The last system, INGEOTEC, corresponds to an approach that takes advantage of the conversation dynamics. It considers the current utterance, the previous, and the next. In the extremes, either the next or the previous are empty utterances. Let $\mathbf{x}$, $\mathbf{x}_p$, and $\mathbf{x}_n$ be the dense representation, then it is computed the similarity between the current utterance ($\mathbf{x}$) and the previous and next utterance, as follows: $s_p = \rho \odot \mathbf{x} \cdot \mathbf{x}_p$, and $s_n = \rho \odot \mathbf{x} \cdot \mathbf{x}_n$. At first, ρ is a vector of ones, so s_p and s_n are the cosine similarity because the dense representations have unit length.

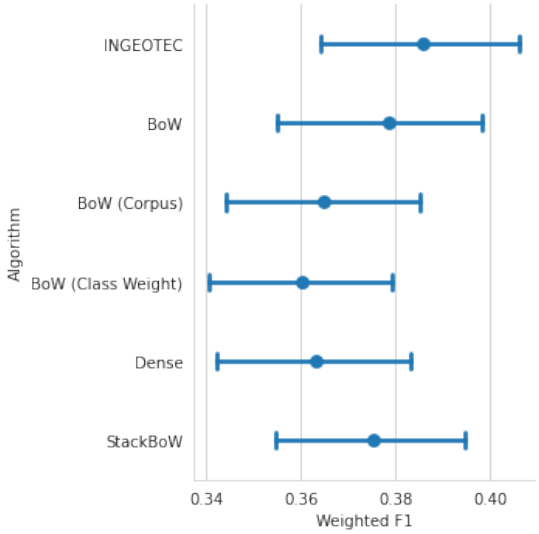
Using s_p and s_n, the contribution of each representation is computed by converting the similarity to a probability; this is done with the softmax as $\mathbf{s} = \mathsf{softmax}(1, s_p, s_n)$. Using $\mathbf{s}$ another representation is created which is the convex combination between $\mathbf{x}$, $\mathbf{x}_p$, and $\mathbf{x}_n$, i.e., $\mathbf{x}_s = \mathbf{s}_1\mathbf{x} + \mathbf{s}_2\mathbf{x}_p + \mathbf{s}_3\mathbf{x}_n$. Using $\mathbf{x}$ and $\mathbf{x}_s$, the dense representation used is the concatenation of them, i.e., $\mathbf{w} = [\mathbf{x}, \mathbf{x}_s]$. The final dense representation, $\mathbf{w}$, is used in a linear equation combined with softmax to predict the probabilities of each class. The probabilities obtained in the previous step are combined, using a convex combination, with the decision function of a BoW classifier (transformed with softmax). It is important to mention that the parameters, ρ, the coefficients to create the final convex combination, and the parameters of the linear equation of the dense representation $\mathbf{w}$ are optimized with gradient descent.

4 Results

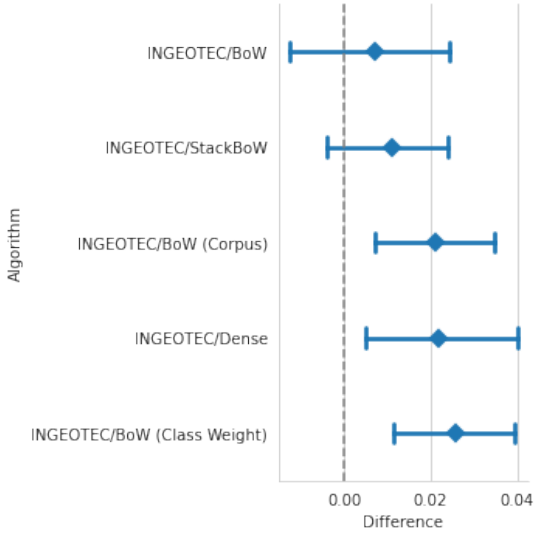
The systems' performance analysis starts with the information presented in Table 1. The table presents the performance, in terms of F1 scores per class, of the BoW (Class Weight) classifier, the Dense classifier, and their convex combination, namely StackBoW. The BoW (Class Weight) classifier is identified using the term *Class Weight* to indicate that the Linear Support Vector Machine was optimized by giving a weight inversely proportional to the class frequencies to each sample. This configuration is also used in the Dense classifier.

Table 1 is organized in three row blocks. The first one identified with the parameter β presents the coefficients used to make the convex combination of BoW and Dense. The second-row block contains the performance (F1 scores per class) estimated in k-fold cross-validation in the training set, and the third block corresponds to the performance in the test set. It can be observed from the table that the performance of StackBoW in the k-fold cross-validation is better than that of its components. This improvement is not reflected in all the cases in the test set; nonetheless, the convex combination is better than its components in the weighted

F1 score.



(a) Weighted F1 score and its estimated confidence intervals (90%) for the different systems.



(b) Difference in performance and its confidence interval (90%) between the best system (namely, INGEOTEC) and the rest.

Figure 1: Analysis of the weighted F1 score in the test set obtained by different algorithms. The dashed line corresponds to zero. An interval crossing the dashed line indicates the difference is not statistically significant with confidence of 90%.

Figure 1 complements the information presented in Table 1. Figure 1a presents the performance using the weighted F1 score on the test set for all the systems tested and its associated 90% confidence interval (the confidence intervals were estimated using the procedure described in (Nava-Muñoz et al., 2023)). The figure also includes the difference in performance (Figure 1b) between the best-performing systems, namely INGEOTEC, and the rest of the systems. The difference in performance shows the 90% confidence interval. The performance of INGEOTEC is 0.3861. The comparison figure includes a dash line that it is set in zero, consequently any confidence interval that intersect with the dash line indicates that the difference in performance is not statistical significant. Using this information, it can be observed that INGEOTEC is similar to the BoW classifier –it is worth mentioning that BoW weights all samples with 1, which makes it different and BoW (Class Weight)– and StackBoW.

5 Conclusion

We have described the algorithms tested on the Emotion Recognition in Conversation task. Most of the approaches treat this problem by looking at each utterance separately. The only system taking advantage of the dynamic of the conversation is the INGEOTEC system; this system is the one having the best performance. Nonetheless, as Figure 1b shows, it is not statistically different than a BoW classifier. The BoW classifier is the simplest model one can start experimenting with.

References

Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. 2022. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3829–3839, Marseille, France. European Language Resources Association.

Mario Graff, Sabino Miranda-Jiménez, Eric S. Tellez, and Daniela Moctezuma. 2020. EvoMSA: A Multilingual Evolutionary Approach for Sentiment Analysis. *Computational Intelligence Magazine*, 15(1):76 – 88.

Mario Graff, Daniela Moctezuma, Eric Tellez, and Sabino Miranda. 2023a. Ingeotec at DA-VINCIS: Bag-of-Words Classifiers. *CEUR Workshop Proceeding*, 3496:1–10.

Mario Graff, Daniela Moctezuma, Eric Tellez, and Sabino Miranda. 2023b. Ingeotec at restmex: Bag-of-words classifiers. *CEUR Workshop Proceedings*, 3496:1–11.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018. ICON: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604, Brussels, Belgium. Association for Computational Linguistics.

Table 1: F1 scores per class, in the training set, using k-fold cross-validation, and in the test set. β corresponds to the value of the convex combination. BoW and Dense are the base classifiers of StackBoW and the predictions of these are combined using β.

	anger	contempt	disgust	fear	joy	neutral	sadness	surprise
β	0.21	0.25	0.96	0.73	0.54	0.35	0.54	0.71
k-fold cross-validation								
BoW (Class Weight)	0.23	0.21	0.14	0.20	0.41	0.52	0.26	0.15
Dense	0.22	0.20	0.12	0.20	0.38	0.46	0.28	0.21
StackBoW	0.24	0.21	0.16	0.23	0.43	0.5	0.29	0.19
Test set								
BoW (Class Weight)	0.23	0.16	0.10	0.18	0.39	0.47	0.24	0.27
Dense	0.25	0.23	0.05	0.13	0.41	0.45	0.32	0.27
StackBoW	0.26	0.22	0.13	0.18	0.44	0.46	0.29	0.27

Dazhi Jiang, Hao Liu, Geng Tu, Runguo Wei, and Erik Cambria. 2024. Self-supervised utterance order prediction for emotion recognition in conversations. *Neurocomputing*, 577:127370.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Yukun Ma, Khanh Linh Nguyen, Frank Z. Xing, and Erik Cambria. 2020. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander F. Gelbukh, and Erik Cambria. 2018. Dialoguernn: An attentive RNN for emotion detection in conversations. *CoRR*, abs/1811.00405.

Sergio Nava-Muñoz, Mario Graff Guerrero, and Hugo Jair Escalante. 2023. Comparison of Classifiers in Challenge Scheme. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13902 LNCS:89–98.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard H. Hovy. 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *CoRR*, abs/1905.02947.

Anvita Saxena, Ashish Khanna, and Deepak Gupta. 2020. Emotion recognition and detection methods: A comprehensive survey. *Journal of Artificial Intelligence and Systems*, 2:53–79.

Yong-Soo Seol, Dong-Joo Kim, and Han-Woo Kim. 2008. Emotion recognition from text using knowledge-based.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. Conceptnet 5.5: An open multilingual graph of general knowledge. *Preprint*, arXiv:1612.03975.

Geng Tu, Jintao Wen, Cheng Liu, Dazhi Jiang, and Erik Cambria. 2022. Context- and sentiment-aware networks for emotion recognition in conversation. *IEEE Transactions on Artificial Intelligence*, 3(5):699–708.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008. Curran Associates, Inc.

Appendix A: Library Usage

This appendix aims to illustrate the use of BoW, Dense, and StackBOW that are implemented using EvoMSA (`evomsa.readthedocs.io`) (Graff et al., 2020). The first step is to install the library, which can be done using the Anaconda package manager with as follows:

```
conda install -c conda-forge EvoMSA
conda install -c conda-forge IngeoML
```

Once EvoMSA is installed, one must load a few libraries.

```
from EvoMSA import BoW, DenseBoW
from EvoMSA.back_prop import StackBoW
from IngeoML.utils import soft_comp_weighted_f1,
    support
```

The BoW classifier is trained with the following instruction; it is assumed that the list D contains as elements dictionaries with two keys: text and klass; the latter is used as the emotion.

```
bow = BoW(lang='hi').fit(D)
```

Let us assume that the test set is in a list of dictionaries, G, where the utterance is in the key text. Then, the following instruction is used to predict the emotion of each utterance.

```
emotions = bow.predict(G)
```

BoW (Class Weight) is initialized with the following instructions.

```
kwargs = dict(dual='auto', class_weight='balanced')
bow = BoW(lang='hi',
          voc_size_exponent=17,
          estimator_kwargs=kwargs).fit(D)
```

On the other hand, the Dense classifier is trained using the following command.

```
kwargs = dict(dual='auto', class_weight='balanced')
dense = DenseBoW(lang='hi',
                 voc_size_exponent=17,
                 estimator_kwargs=kwargs).fit(D)
```

The StackBoW classifier is trained with the next step. Finally, all the classifiers have the method *predict* to forecast the emotions of any given utterance.

```
kwargs = dict(class_weight=support)
stack_bp = StackBoW(lang='hi',
                    deviation=soft_comp_weighted_f1,
                    voc_size_exponent=17,
                    optimizer_kwargs=kwargs).fit(D)
```

IIMAS at SemEval-2024 Task 9: A Comparative Approach for Brainteaser Solutions

Cecilia Reyes-Peña
IIMAS/ México
UPMH/ México.
ceciliareyes
@turing.iimas.unam.mx

Orlando Ramos-Flores
IIMAS / México
orlando.ramos
@aries.iimas.unam.mx

Diego Martínez-Maqueda
UPMH / México
231220009@upmh.edu.mx

Abstract

In this document, we detail our participation experience in SemEval-2024 Task 9: BRAINTEASER-A Novel Task Defying Common Sense. We tackled this challenge by applying fine-tuning techniques with pre-trained models (BERT and RoBERTa Winogrande), while also augmenting the dataset with the LLMs ChatGPT and Gemini. We achieved an accuracy of 0.93 with our best model, along with an F1 score of 0.87 for the Entailment class, 0.94 for the Contradiction class, and 0.96 for the Neutral class.

1 Introduction

The brainteasers are problems or puzzles, typically designed to be solved for amusement. To solve brainteasers is necessary the lateral and vertical think, so interpret the context itself contained in them. The SemEval 2024 BRAINTEASER: A Novel Task Defying Common Sense task poses a set of brainteasers and their answers, divided into two types: Sentence Puzzles and Word Puzzles, both in the English language and require an understanding of common sense and the ability to overwrite them through unconventional thinking that distinguishes these defaults from fixed constraints. In Sentence Puzzles, a challenge is presented that defies common sense focused on sentence fragments. In Word Puzzles, the answer challenges the predefined meaning of the word and focuses on the letter composition of the target question (Jiang et al., 2024).

Solving brainteasers requires an unconventional or out-of-the-box approach, which stimulates lateral thinking. This style of thinking is crucial for discovering ingenious solutions to complex problems and for considering situations from multiple perspectives. This type of thinking must be integrated into language models, as it enables them to provide diverse perspectives and apply them to more complex aspects of language, such as understanding metaphors, idioms, or ambiguities.

This paper documents the participation of the IIMAS team at SemEval-2024 task 9, where the resolution of brainteasers was approached using a classification framework. Our strategy relied on fine-tuning techniques applied to pre-trained models using a transformer architecture. In addition to describing our approach, we also analyze the challenges encountered during the process and discuss potential areas for improvement in future research. This paper sheds light on the application of cutting-edge techniques in natural language processing to tackle comprehension and reasoning problems, such as brainteasers, and provides valuable insight into the performance and limitations of our approach in this specific context. During the evaluation phase, the results placed us at the 33th out of 50 participants.

2 Background

We examine various methodologies for solving brain teaser challenges. In this overview, we present some of these approaches. Mitra and Baral (2015) focused on solving logic grid puzzles. Initially, they identified keywords as entities and the relationships between them. Subsequently, they constructed a pair of Answer Set Programming rules. These rules served as inputs for a logic reasoner named Logicia, equipped with a predefined set of predicates. Their model demonstrated an impressive 85.05% accuracy in classifying constituents and successfully solved 71 out of 100 test puzzles. The RIDDLESENSE challenge, introduced by Lin et al. (2021), aims to explore the task of answering riddles. This challenge presents participants with a multiple-choice question-answering scenario, where a model must select one answer from a set of five choices (one correct answer and four distractors) in response

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1121–1126

to a given riddle question. The dataset comprises 5.7k meticulously curated examples. In their experiments, researchers employed various approaches including fine-tuning pre-trained language models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020), alongside fine-tuning a text-to-text QA Model (Khashabi et al., 2020). Their methodology involved concatenating the question with the answer choices. During evaluation, three native English speakers achieved an average accuracy of 91.3%, with the best-performing model achieving 68.8% accuracy.

Current language models can be evaluated in what is known as vertical or convergent thinking and perform well; however, the existence of lateral or divergent thinking in the human mind leads to considering the option of evaluating these same models in this way of thinking. This idea is taken by Huang et al. (2023) to propose a way to evaluate Large Language Models (LLMs) in Lateral Thinking Puzzles, also known as situations puzzles. This type of puzzle involves a host who knows the complete truth but gives the player a story lacking certain information. The player, through questions that are only answered with Yes or No, must deduce the whole truth. The GTP-4 model from OpenAI had the best performance in this type of puzzle according to the proposed evaluation.

Tong et al. (2023) also identified the need for non-linear thinking in LLMs, so in their work, they proposed Inferential Exclusion Prompting (IEP) inspired by the method of elimination thinking. This proposal consists of, given a problem, the IEP instructs the LLMs to plan different responses and then eliminate those options that are contradictory or irrelevant. The IEP was evaluated for various problems: parajumbles, riddles, puzzles, brain teasers, and critical reasoning queries against Chain-of-Thought (CoT) prompting.

3 System overview

The data used in this task were provided by Jiang et al. (2023), comprising a set of 507 brainteasers for sentence puzzles and 396 brainteasers for word puzzles. Each of these brainteasers includes one correct answer alongside three distractors. This dataset showcases the complexity of the posed problems, suggesting that they can be effectively addressed through a natural language understanding (NLI) approach.

In this context, the BART model (Lewis et al., 2019) serves as an option for resolving Multi-Genre Natural Language Inference (MultiNLI) problems, where a model's ability to determine which of the proposed premises is true relative to a hypothesis is evaluated using a multi-choice approach. We apply zero-shot classification to the BART model, and as result, we got a low performance as we describe in Table 1.

Table 1: Multi-choice approach accuracy.

Data	Accuracy
SP-train	0.2879
WP-train	0.2449

Given the suboptimal performance of zero-shot models in multichoice tasks, the decision was made to fine-tune a model. One initially discarded proposal was to utilize the MultiNLI dataset (Williams et al., 2018)[1] for fine-tuning, as the BART model[2] is trained on this data and yielded unsatisfactory results. Therefore, the decision was made to work with data provided by the competition or data sharing of a similar nature.

To accomplish this, data transformation was necessary to operate under a classification approach, where each question serves as a value for the premise feature, and each answer is treated as a value for the hypothesis feature, these being the indicators: distractor1, distractor2, distractor(unsure), and correct answer. Each pair of data is assigned a class label. For fine-tuning bert-base-uncased, three different classes are managed. For sentence pairs containing distractor1 and distractor2, the corresponding label is Contradiction; for distractor(unsure), it is Neutral, and for the correct answer, it is Entailment (see Fig 1). For the RoBERTa Winogrande model Sakaguchi et al. (2019), it is expected that the resulting sentence from concatenating the premise with the hypothesis will have a boolean value depending on the dependencies of the hypothesis concerning the premise. Therefore, the labels used are False and True. Both models utilize the following hyperparameter values: batch_size=32, epochs=3, learning_rate=2e-5, as well as a split of the dataset with 80% for training and 20% for evaluation purposes.

In order to enhance the performance of the models, we leveraged the unique capabilities of large language models (LLMs). We employed ChatGPT

[1]https://huggingface.co/datasets/multi_nli
[2]https://huggingface.co/facebook/bart-large-mnli

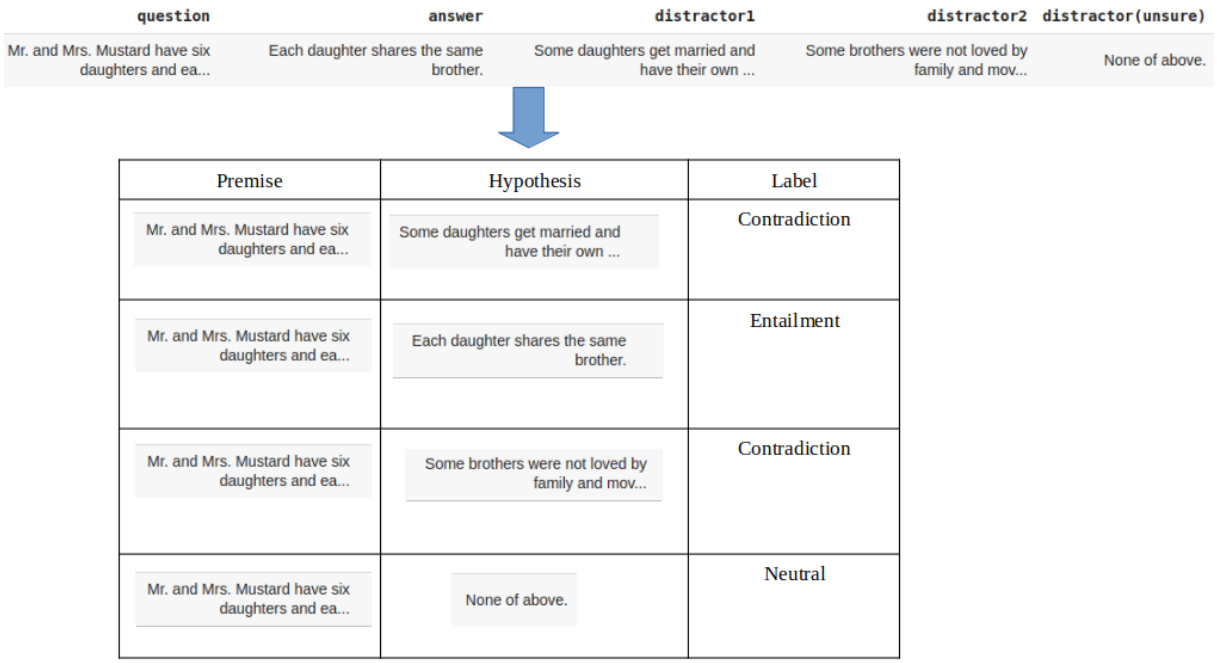


Figure 1: Data Transformation for BERT Classification Approach.

3.5[3] and Gemini[4] to generate additional brainteaser instances. These instances were then incorporated into the fine-tuning process of pre-trained models. Despite having more examples due to data transformation, additional examples were generated through language models such as ChatGPT and Gemini. The generated data underwent manual review to prevent errors regarding the correct answers to the brainteasers. With the expansion and transformation of the data, a total of 4,644 labeled pairs were obtained for fine-tuning the models with brainteasers from both tasks.

4 Experimental Setup

The evaluation results of the BERT Fine-Tuning model are presented in Table 2, revealing the model's struggle to identify the correct answer while being proficient in identifying the neutral class. Based on these findings, a decision was made to minimize the dataset size, considering the potential for model overfitting.

Consequently, the use of brainteasers generated for the Word Puzzle task was discarded, as is shown in Table 3. This decision impacted the model's performance, as evidenced in Table 4, prompting further reduction of the training dataset.

After eliminating all synthetically generated

Table 2: Evaluation Metrics of BERT Fine-Tuning Model with Original Data Train and Generated Data for Sentence Puzzle and Word Puzzle Tasks (Model 1).

Class	Precision	Recall	F1-score
Entailment	0.80	0.78	0.79
Contradiction	0.90	0.91	0.90
Neutral	**0.96**	**0.97**	**0.97**
Macro avg	0.89	0.89	0.89
Weighted avg	0.89	0.89	0.89
Accuracy			0.89

Table 3: Evaluation Metrics of BERT Fine-Tuning Model with Original Data Train and Generated Data for Sentence Puzzle task (Model 2).

Class	Precision	Recall	F1-score
Entailment	0.90	0.79	0.84
Contradiction	0.91	0.96	0.93
Neutral	**0.96**	**0.98**	**0.97**
Macro avg	0.92	0.91	0.91
Weighted avg	0.92	0.92	0.92
Accuracy			0.92

[3]https://chat.openai.com/
[4]https://gemini.google.com/

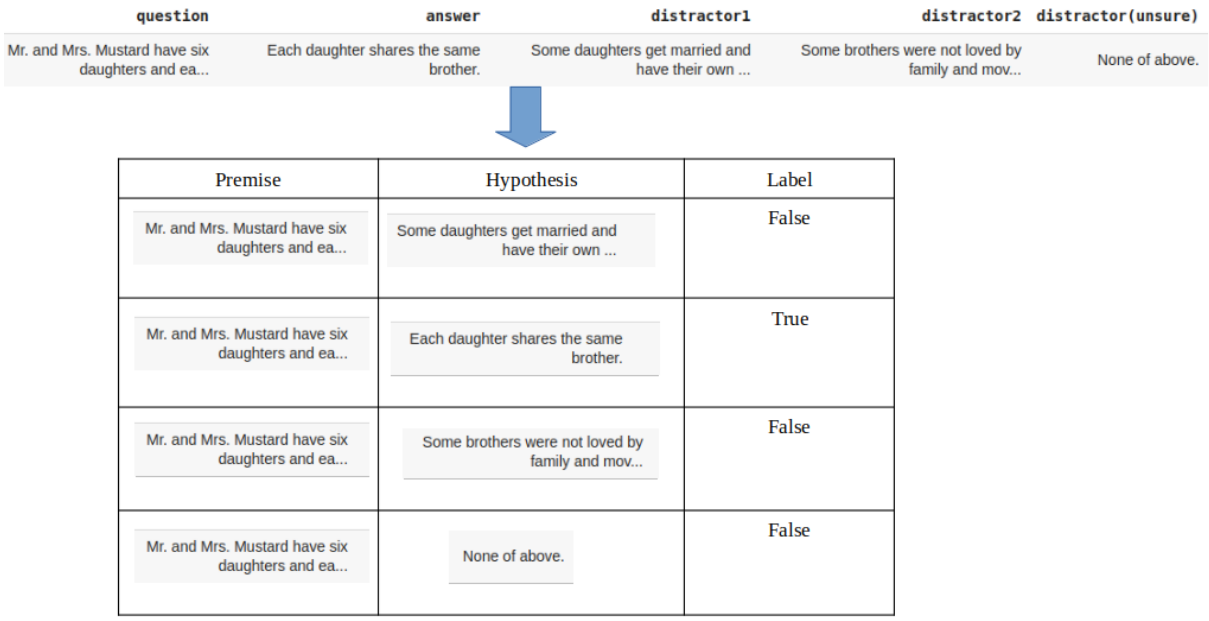


Figure 2: Data Transformation for RoBERTa Winogrande Classification Approach.

data, a noticeable improvement in the evaluation metrics for *Entailment* and *Contradiction* classes was achieved, as we present in Table 4.

Table 4: Evaluation Metrics of BERT Fine-Tuning Model with Original Data Train only for Sentence Puzzle task (Model 3).

Class	Precision	Recall	F1-score
Entailment	0.91	0.84	0.87
Contradiction	0.93	0.96	0.94
Neutral	**0.95**	**0.97**	**0.96**
Macro avg	0.93	0.92	0.92
Weighted avg	0.93	0.93	0.93
Accuracy			0.93

Finally, with the selected data in hand and the pursuit of further improvement, fine-tuning of the RoBERTa Winogrande model was carried out. However, the results were not comparable to those obtained during the fine-tuning of BERT, leading to the decision to discard this model (see Table 5).

Table 5: Evaluation model metrics.

Class	Precision	Recall	F1-score
False	0.73	1	0.84
True	0.0	0.0	0.0
Macro avg	0.36	0.50	0.42
Weighted avg	0.53	0.73	0.61
Accuracy			0.73

Table 6 displays the results obtained during the training stage using the evaluation metrics proposed for the task. For the evaluation phase, Model 3 was selected as it exhibited the best performance.

5 Result

During the evaluation phase of SemEval-2024 Task 9, administrators provided a dataset comprising 120 sentence puzzles and 96 word puzzles. The results, depicted in Table 6, demonstrate that the majority of these results surpass the baseline established by the zero-shot models. Our average final ranking, as displayed in the posted rankings table, is 33, with a score of 0.658 for Sentence Puzzle (position 23) and 0.260 for Word Puzzle (position 22), yielding an overall average score of 0.459.

5.1 Error Analysis

The primary errors of the proposed algorithm are associated with the word puzzle task, as evidenced by the imbalance of classes. Despite efforts to mitigate this imbalance by generating additional data, addressing this task has proven challenging, as the results did not exhibit improvement. One possible contributing factor to this challenge is the necessity for a deeper contextual understanding and a more figurative sense to solve these puzzles.

6 Conclusions

This work introduced a solution for SemEval-2024 Task 9: "BRAINTEASER - A Novel Task Defying Common Sense", leveraging pre-trained lan-

Table 6: SemEval2024 Task 9: BRAINTEASER train data results

	Sentence Puzzle						Word Puzzle					
Train set	**Original**	**Semantic**	**Context**	**Orig. + Sem.**	**Orig. + Sem. + Con.**	**Overall**	**Original**	**Semantic**	**Context**	**Orig. + Sem.**	**Orig. + Sem. + Con.**	**Overall**
Bard zero-shot	.284	.289	.289	.224	.13	.243	.189	.265	.28	.174	.068	.195
Model 1	.81	.828	.721	.81	.692	.77	.174	.181	.136	.09	.037	.123
Model 2	.887	.893	.846	.881	.822	.865	.272	.257	.28	.113	.03	.19
Model 3	.911	.911	.863	.911	.863	.891	.212	.174	.212	.19	.037	.145

Table 7: SemEval2024 Task 9: BRAINTEASER results table

	Sentence Puzzle						Word Puzzle					
Test set	**Original**	**Semantic**	**Context**	**Orig. + Sem.**	**Orig. + Sem. + Con.**	**Overall**	**Original**	**Semantic**	**Context**	**Orig. + Sem.**	**Orig. + Sem. + Con.**	**Overall**
Human	.907	.907	944	.907	.889	.920	.917	.917	.917	.917	.900	.917
ChatGPT	.608	.593	.679	.507	.397	.627	.561	.524	.518	.439	.292	.535
RoBERTa-L	.435	.402	.464	.330	.201	.434	.195	.195	.232	.146	.061	.207
IIMAS Team	.65	.675	.650	.600	.500	.658	.250	.250	.281	.125	.062	.260

guage models and fine-tuning them with the provided data (Jiang et al., 2023), along with additional data generated using LLMs as ChatGPT 3.5 and Gemini. Through experimentation with our pre-trained, fine-tuned models, we found that the BERT model yielded the best results compared to RoBERTa Winogrande. It is worth noting that a significant challenge in this process was defining the appropriate dataset, as certain records from the proposed set had to be discarded to enhance model performance. Ultimately, our results surpassed the task's baseline and secured a position of 33 out of 50 participants, indicating the effectiveness of our approach. However, there is room for improvement, particularly with the word puzzles, which proved to be challenging and require a deeper contextual understanding for resolution.

Acknowledgements

The authors thankfully acknowledge the computer resources, technical expertise and support provided by the Laboratorio Nacional de Supercómputo del Sureste de México, CONACYT member of the network of national laboratories. We also thank to Consejo de Ciencia, Tecnología e Innovación de Hidalgo for the support provided in the program *"Becas para Posgrados de Excelencia, Maestría y Doctorado"*.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, and Hai-Tao Zheng. 2023. Lateval: An interactive llms evaluation

benchmark with incomplete information from lateral thinking puzzles.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1504–1515.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Arindam Mitra and Chitta Baral. 2015. Learning to automatically solve logic grid puzzles. In *Conference Proceedings - EMNLP 2015*, Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, pages 1023–1033. Association for Computational Linguistics (ACL). Publisher Copyright: © 2015 Association for Computational Linguistics.; Conference on Empirical Methods in Natural Language Processing, EMNLP 2015 ; Conference date: 17-09-2015 Through 21-09-2015.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Yongqi Tong, Yifan Wang, Dawei Li, Sizhe Wang, Zi Lin, Simeng Han, and Jingbo Shang. 2023. Eliminating reasoning via inferring with planning: A new framework to guide llms' non-linear thinking.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

PetKaz at SemEval-2024 Task 3: Advancing Emotion Classification with an LLM for Emotion-Cause Pair Extraction in Conversations

Roman Kazakov, Kseniia Petukhova, Ekaterina Kochmar
Mohamed bin Zayed University of Artificial Intelligence
{roman.kazakov, kseniia.petukhova, ekaterina.kochmar}@mbzuai.ac.ae

Abstract

In this paper, we present our submission to the SemEval-2023 Task 3 "The Competition of Multimodal Emotion Cause Analysis in Conversations", focusing on extracting emotion-cause pairs from dialogs. Specifically, our approach relies on combining fine-tuned `GPT-3.5` for emotion classification and a BiLSTM-based neural network to detect causes. We score 2nd in the ranking for Subtask 1, demonstrating the effectiveness of our approach through one of the highest weighted-average proportional F_1 scores recorded at 0.264. Our code is available at https://github.com/sachertort/petkaz-semeval-ecac.

1 Introduction

Developing dialog systems is a complex task that has attracted considerable attention from many technology companies and universities over the last 70 years since the introduction of Eliza in 1966 (Weizenbaum, 1966). Modern large language models (LLMs) like GPT-4 (OpenAI, 2023) are trained to avoid causing harm and often assert their lack of personal opinions on intricate matters, which is not at all natural for conversations. They do not respond in a way that is truly human, and they do not understand the range of feelings that words can cause. Recognizing the emotional implications of an utterance provides a deeper understanding of dialog, enabling the development of more human-like dialog systems. These systems could navigate conversations using a comprehensive understanding of emotional dynamics and planning responses based on this understanding rather than just predicting likely outcomes.

To bridge the gap between machine-generated dialogs and rich, complex human communication, we develop models for SemEval-2024 Task 3 "The Competition of Multimodal Emotion Cause Analysis in Conversations"[1] (ECAC) (Wang et al., 2024). This task was previously introduced in Xia and Ding (2019a) and later in Wang et al. (2023), where the authors also described a multimodal dataset called *Emotion-Cause-in-Friends* (ECF) for this task.

We focus only on Subtask 1, "Textual Emotion-Cause Pair Extraction in Conversations" (ECPE),[2] where the goal is to classify emotions and extract the corresponding textual causal spans. To accomplish this, we propose a two-stage pipeline: (1) first, emotions are classified using a fine-tuned LLM, and then (2) causes are extracted with a simple neural network consisting of BiLSTM and linear layers (see Figure 1). Our system achieved a weighted-average proportional F_1 score of 0.264, the primary metric in this competition's evaluation phase on the test set. Consequently, our team ranked 2nd out of 15 participating teams based on this metric. We provide an extensive analysis of the model's performance in Section 5.2.

2 Related Work

Recent research in the field of dialog systems and emotion-cause extraction has seen significant advancements through various innovative approaches, some of which we overview in this section. For instance, Chen et al. (2023) introduce a novel technique that uses graphs to model "causal skeletons" alongside a causal autoencoder (CAE) for refining these models by integrating both implicit and explicit causes.

Following closely, Zhang et al. (2023) present Dual Graph Attention Networks (DualGATs) that leverage discourse structure and speaker context through a combination of Discourse-aware GAT (DisGAT) and Speaker-aware GAT (SpkGAT), enriched with an interaction module for effective information exchange and context capturing.

Moving to earlier work, Kong et al. (2022) pro-

[1] https://nustm.github.io/SemEval-2024_ECAC/

[2] We did not participate in the multimodal track.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1127–1134

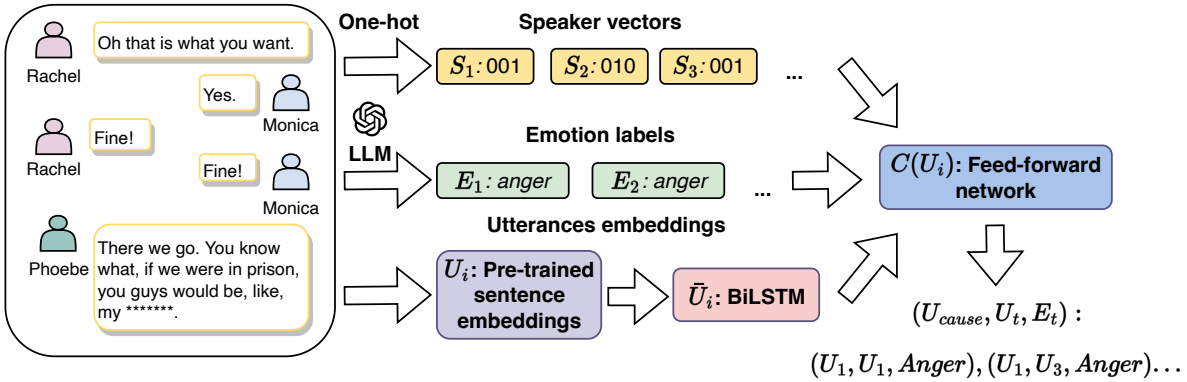


Figure 1: The pipeline for ECPE. Utterances are classified with emotion labels E_i, and speakers are represented with one-hot vectors S_i. Utterances are then encoded with pre-trained sentence embeddings U_i and enriched with context by BiLSTM $\bar{U}_i$. For each target utterance U_t, we detect whether any other utterance from the conversation history $H(U_t)$ is causal using a feed-forward network. $\bar{U}_i$, S_i (of a potential causal utterance), $\bar{U}_t$, S_t, and E_t are concatenated, and then, binary classification is performed. The pipeline outputs labelled emotion-cause pairs (U_i, U_t, E_t).

pose a discourse-aware model (DAM) that integrates emotion cause extraction with discourse parsing, using a Gated Graph Neural Network (GNN) to encode discourse structures and conversation features within a multi-task learning framework, enhancing the understanding of conversational context and structure.

Finally, Gao et al. (2021) focus on improving dialog systems' empathetic response generation by identifying emotion causes. Their framework combines an emotion reasoner for predicting emotion and its cause with a response generator that employs a gated attention mechanism to emphasize important words, exploring both hard and soft gating strategies.

3 System Overview

Our pipeline consists of two stages. Specifically, to identify emotion-cause pairs and emotion types, dialogs are passed through the following modules:

1. classification of utterances with emotion types (including *neutral* for non-emotional utterances) with a supervised fine-tuned LLM; and

2. extraction of cause utterances with a BiLSTM-based network.

The full pipeline is shown in Figure 1. Due to the limitations of the data, we perform the tasks separately, and we elaborate on each of the stages in the following sections.

3.1 Emotion classification

To categorize an utterance with an emotion label E_t, within our pipeline an LLM should consider both the target utterance U_t, which is the t^{th} utterance in a conversation, and the preceding utterance U_{t-1}. It is especially important when we deal with very short turns, such as "Instead of... ?", "No.", "Yeah, maybe...". Indeed, it would be more accurate to utilize causal utterances rather than antecedent ones; however, at the initial stage, these are unknown to us, necessitating the use of a meaningful alternative.

For this stage, we fine-tune `GPT-3.5`.[3] As a `system`'s input, we provide the prompt consisting of an instruction, U_{t-1} (`<UTT_1>`), and U_t (`<UTT_2>`) as is shown in Figure 2. This particular prompt was selected during the preliminary prompt engineering stage. The `assistant`'s output consists of one word – the emotion type.

We also note that preliminary experiments showed that the LLM performed poorly in zero-shot and few-shot settings on the emotion detection task, at least on the ECF dataset (see Section 5.1 and Table 2). Therefore, we had to fine-tune it.

3.2 Cause extraction

The second stage is concerned with the detection of the causal utterances for non-emotional utterances in a binary way. Let the whole conversational history of an utterance U_t be $H(U_t) = [U_1, U_t]$, then the set of all causal utterances is $C(U_t) \subseteq H(U_t)$.

[3] `gpt-3.5-turbo-1106`: https://platform.openai.com/docs/models/gpt-3-5-turbo

```
Take a deep breath. Your task: given two
dialog utterances, predict an emotion of
the second utterance. Select the emotion
from the following options: neutral,
anger, disgust, fear, joy, sadness,
surprise. Do not use any other
emotions!!! Respond only with the chosen
emotion, without any additional
explanation. Remember that you can only
use listed emotions!!!

Utterance 1: <UTT_1>
Utterance 2: <UTT_2>

Emotion:
```

Figure 2: The prompt used to perform emotion classification with `GPT-3.5`.

In addition, speakers are encoded with one-hot vectors $S_1 ... S_n$ within each dialog.

First, we need to enrich utterance embeddings $U_1 ... U_n$[4] obtained from a pre-trained model with the context within the conversation. Bidirectional LSTM (Hochreiter and Schmidhuber, 1997) was chosen because it can preserve context information in sequential settings using the content of the previous hidden state in encoding the current one. This way, we get new utterance representations $\bar{U}_1 ... \bar{U}_n$.

Then, for each target utterance U_t with $E_t \neq neutral$, we construct t representations:

$$\bar{U}_i \parallel S_i \parallel \bar{U}_t \parallel S_t \parallel E_t, \forall i \in [1, ..., t] \quad (1)$$

containing one of the previous utterances or the target utterance embedding itself $\bar{U}_i$ as a potential cause, its speaker vector S_i, the target utterance embedding $\bar{U}_t$, its speaker vector S_t, and the emotion label E_t. We pass them to a feed-forward neural network and obtain binary predictions $\{0, 1\}$, where 1 means that U_i is a causal utterance and 0 stands for the opposite. All U_i for which 1 is predicted make up $C(U_t)$. Thus, for each U_t with $E_t \neq neutral$ we obtain from 0 to t labelled emotion-cause pairs (U_i, U_t, E_t), where $U_i \in C(U_t)$, consisting of the causal utterance,[5] the emotion utterance, and the emotion label.

We have decided not to extract specific spans from the utterances classified as causes, following a thorough review of the dataset. This decision is based on our observation that these spans often defy straightforward explanations, even from a human annotator perspective. Here are some examples, where the rationale behind the spans remains unclear to us:

- The final punctuation marks are often not included in the cause span: e.g., while the complete utterance is *Instead of [...]?*, the identified cause span is *Instead of [...]*
- For the statement *Me, I ... I went for the watch*, the span is *I went for the watch*
- For the sentence *You know you probably did not know this, but back in high school, I had a, um, major crush on you*, the cause span is defined as *you probably did not know this, but back in high school, I had a, um, major crush on you*

We believe that this part of the task can be more accurately defined as a causal emotion entailment (Poria et al., 2021). Additionally, we note that there is an inconsistency in the dataset's annotation: specifically, the task organizers define emotion causes by identifying specific spans within an utterance, yet the emotional responses themselves are treated as consisting of entire utterances. For these reasons, we have decided that it would be methodologically more appropriate to omit the exact span detection step from our pipeline.

4 Experimental Setup

4.1 Data

The dataset proposed for the shared task contains conversations from the *Friends* series annotated with emotion-cause pairs and emotion labels, including *anger*, *disgust*, *fear*, *joy*, *sadness*, *surprise* from Ekman et al. (1987), or *neutral* for non-emotional utterances.

The shared task organizers highlight that 91% of emotions have corresponding causes and one emotion may be triggered by multiple causes in different utterances. In addition, we have noticed that 16% of them cause several different emotions.

The organizers did not provide a standalone development set, so we had to split the training set ourselves using a ratio of 9:1 relative to the dialogs. The final data splits are shown in Table 1.

[4] We use the same notation for utterances and their embeddings for simplification purposes.

[5] We did not extract causal spans and used the whole causal utterance in the evaluation.

Set	# dialogs	# utterances	# EC
Training	1,236	12,346	8,565
Development	138	1,273	799
Total	1,374	13,619	9,364

Table 1: Distribution of dialogs, utterances, and emotion-cause pairs ("EC") across the split sets.

4.2 Training and architecture details

We fine-tune `GPT-3.5` with the default hyperparameters recommended by OpenAI[6] using two epochs, which is the number automatically chosen by the platform.

The cause extractor model is initialized with mean pooling from the penultimate layer's hidden state of the pre-trained `bert-base-uncased`.[7] Our neural network consists of three BiLSTM layers, one hidden linear layer accompanied by batch normalization, and a ReLU activation function.

For training, we employ the Adam optimizer with the learning rate of $1e$-4, weight decay (L_2-norm regularization) of $1e$-5, and cross-entropy as the loss function. We train the model for 200 epochs using a batch size of 32.

As a framework for training and evaluation, we use `PyTorch`[8] (Paszke et al., 2019).

4.3 Evaulation measures

As proposed in the shared task, we apply the weighted average (w-avg.) F_1 score by emotion type for evaluation. The specific implementation of F_1 score for the ECPE task can be found in Xia and Ding (2019b). In this setting, an emotion-cause pair is considered as correctly predicted if the index of an emotion utterance, an emotion type, and the index of the cause utterance match the entry in the gold dataset. There are two strategies related to causal span detection: *strict* F_1 (the same span) and *proportional* F_1 (overlap).[9]

5 Results

Our final submission was evaluated on the test set and achieved the following results:

- w-avg. proportional F_1: 0.264;
- w-avg. strict F_1: 0.104.

[6] https://platform.openai.com/docs/api-reference/fine-tuning/create

[7] https://huggingface.co/bert-base-uncased

[8] https://pytorch.org

[9] For the details on the metrics, refer to https://github.com/NUSTM/SemEval-2024_ECAC/tree/main/CodaLab/evaluation.

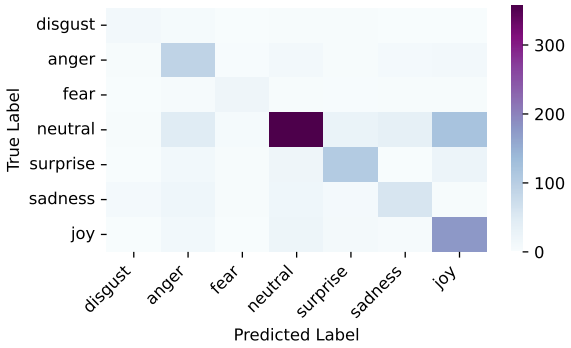


Figure 3: Performance of our emotion classifier on our development set.

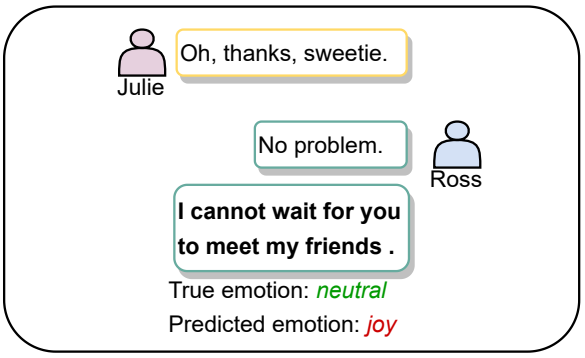


Figure 4: An example of a dialog where our model classified neutral utterance as *joy*.

As a result, we score second out of fifteen teams participating in Subtask 1 according to the main shared task metric – w-avg. proportional F_1.

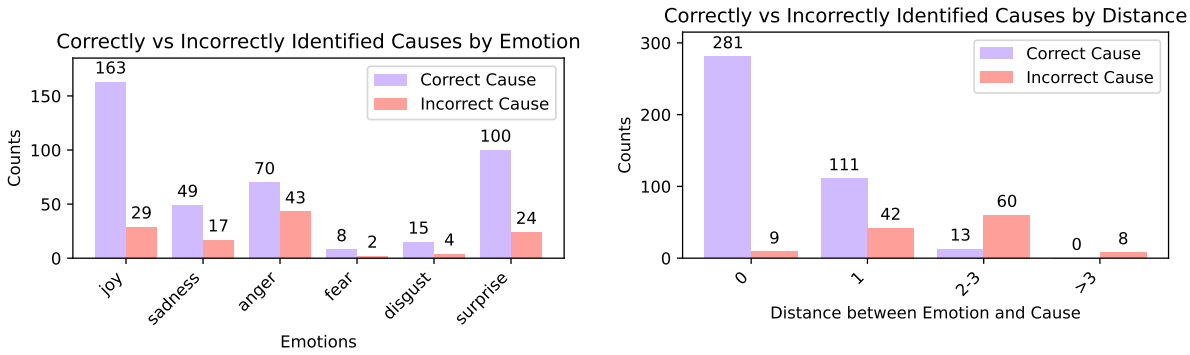
5.1 Emotion classification performance

Table 2 overviews the performance of emotion classification using `GPT-3.5` across different paradigms: zero-shot, few-shot, and fine-tuning. We note that zero- and few-shot settings use the same prompt (see Figure 2), with the few-shot setting including one handpicked example per each emotion type (see Appendix A). As expected, fine-tuning yields the best results on all emotion types and overall. Interestingly, few-shot prompting performs worse than zero-shot, which suggests that examples hamper the model's understanding of emotion types instead of improving it.

Utterances of *disgust* type turn out to be the most difficult to predict correctly: one of the possible reasons is that they are insufficiently represented in the training set (amounting to only about 6% of emotional utterances). However, the zero-shot and few-shot settings also show the poorest performance on *disgust*.

Approach	*neutral*	*anger*	*disgust*	*fear*	*joy*	*sadness*	*surprise*	macro	w-avg.
Zero-shot	0.61	0.43	0.30	0.32	0.54	0.47	0.50	0.45	0.54
Few-shot	0.57	0.49	0.31	0.34	0.54	0.37	0.41	0.43	0.51
Fine-tuning	**0.70**	**0.57**	**0.42**	**0.51**	**0.63**	**0.52**	**0.66**	**0.57**	**0.64**

Table 2: F_1 scores on emotion classification with GPT-3.5 across different approaches.



(a) Analysis across emotions on our development set (on correctly identified emotions only).

(b) Break-down of distance between emotion and cause on our development set (on correctly identified emotions only).

Figure 5: Performance of the cause extractor.

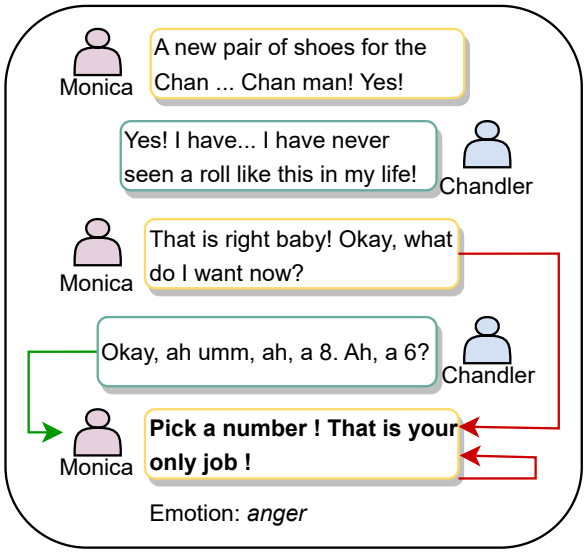


Figure 6: An example of a dialog with annotated causes for *anger* (green for causes correctly identified by our model, and red for causes that our model failed to recognize).

Our analysis of the emotion classifier's performance across different emotion types shows that the model often incorrectly classifies *neutral* utterances as indicative of *joy* (see Figure 3). After further investigation, we have found that a large number of these incorrectly categorized cases contain greetings ("*Hi!*") and expressions of gratitude ("*Thank you!*", "*You're welcome!*), which, according to our dataset, should be *neutral*, yet our classifier interprets them as *joy*. This implies that text alone may not be enough to identify an emotion, given that such utterances can express joy or remain emotionally neutral. There are other controversial cases, such as a conversation between two lovers shown in Figure 4, where the statement "*I cannot wait for you to meet my friends*" is actually more likely to express joy rather than neutrality.

5.2 Analysis

We also evaluate our model on its ability to identify the causes of utterances expressing different emotions, as shown in Figure 5a. Based on this analysis, the greatest challenge for our model is determining causes of *anger*. Similarly, manual analysis shows that this task is difficult for humans as well. As an example, Figure 6 highlights a scenario where the source of anger in Monica's utterance is not only attributed to the preceding utterance from Chandler but is also caused by the utterance that came before Chandler's, as well as the context of Monica's own statement. Intricacies like this one highlight the controversies present in the dataset.

Additionally, we have looked into how well our model performs in determining the emotion's cause based on how close it is to the emotional utterance, as we show in Figure 5b. First of all, it transpires that most emotional utterances are self-caused. Furthermore, our analysis shows that there is a clear correlation between the cause's distance from the

emotional utterance and our model's identification accuracy: the further away the cause, the lower the model's performance.

In the course of our analysis, we have discovered instances where emotions appear before their causes. This observation suggests that the organizers' definition of a cause in dialog contexts is non-trivial, as, typically, we would expect that something happens and triggers an emotion. However, in the case of the preceding emotion, the cause is fundamentally different: it is a reason in terms of linguistics and it explains the emotion, but it does not trigger it (for the difference between CAUSE and REASON, please refer to Ruppenhofer et al., 2006).

Overall, accurate identification of emotions and their causes within utterances proves to be a complex challenge, not only for models but also for humans. All issues mentioned above point to important problems in the dataset that need to be carefully thought through and fixed to enhance both the accuracy and reliability of ECPE efforts.

6 Conclusions

Our work presents a novel approach to emotion-cause pair extraction in conversations, using the capabilities of an LLM (specifically, `GPT-3.5`) for emotion classification. This methodology is further enhanced by the use of a BiLSTM-based neural network for extracting causes. Our system outperforms most of the submissions to the shared task, scoring 2nd in the overall ranking according to the main metric of weighted-average proportional F_1. For future enhancements to our pipeline, we consider the following improvements:

- Firstly, data annotation itself can be expanded and improved, potentially via the use of an LLM for annotation.
- Secondly, speaker representations can be improved to enhance the understanding and processing of the dialogs.
- Finally, more accurate methods of LLM-based cause extraction can be developed further.

Limitations

Due to OpenAI's policy, we are unable to share our fine-tuned model. Therefore, those wishing to reproduce our experiments will need to do the fine-tuning independently. Overall, the usage of an open-source solution instead of a proprietary LLM can be one of the future directions. Also, it may be applied using a more specific framework like InstructERC (Lei et al., 2024).

Furthermore, our research is limited to the emotions present in the provided task data. Consequently, adding new emotions would require further fine-tuning. Due to the shared task rules, we have to develop our system based only on the presented dataset that is limited to a single concrete domain (*Friends* series) and the English language.

Acknowledgements

We are grateful to Mohamed bin Zayed University of Artificial Intelligence (MBZUAI) for supporting this work. We also thank the anonymous reviewers for their valuable feedback.

References

Hang Chen, Xinyu Yang, Jing Luo, and Wenjing Zhu. 2023. How to enhance causal discrimination of utterances: A case on affective reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 494–512, Singapore. Association for Computational Linguistics.

Paul Ekman, Wallace Friesen, Maureen O'Sullivan, A. Chan, Irene Diacoyanni-Tarlatzis, Karl Heider, Rainer Krause, William LeCompte, Tom Pitcairn, and Pio Ricci Bitti. 1987. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology*, 53:712–7.

Jun Gao, Yuhan Liu, Haolin Deng, Wei Wang, Yu Cao, Jiachen Du, and Ruifeng Xu. 2021. Improving empathetic response generation by recognizing emotion cause in conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 807–819, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Dexin Kong, Nan Yu, Yun Yuan, Guohong Fu, and Chen Gong. 2022. Discourse-aware emotion cause extraction in conversations. *arXiv preprint arXiv:2210.14419*.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2024. InstructERC: Reforming emotion recognition in conversation with a retrieval multi-task LLMs framework.

OpenAI. 2023. GPT-4 technical report.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor

Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, et al. 2021. Recognizing emotion cause in conversations. *Cognitive Computation*, 13:1317–1332.

Josef Ruppenhofer, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. *FrameNet II: Extended Theory and Practice*. International Computer Science Institute, Berkeley, California. Distributed with the FrameNet data.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Joseph Weizenbaum. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Rui Xia and Zixiang Ding. 2019a. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Rui Xia and Zixiang Ding. 2019b. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

Duzhen Zhang, Feilong Chen, and Xiuyi Chen. 2023. DualGATs: Dual graph attention networks for emotion recognition in conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7395–7408, Toronto, Canada. Association for Computational Linguistics.

A Prompt for Few-shot

```
Take a deep breath. Your task: given two dialog utterances, predict an emotion of the second
utterance. Select the emotion from the following options: neutral, anger, disgust, fear, joy,
sadness, surprise. Do not use any other emotions!!! Respond only with the chosen emotion,
without any additional explanation. Remember that you can only use listed emotions!!!

Examples:

Utterance 1: Alright , so I am back in high school , I am standing in the middle of the
cafeteria , and I realize I am totally naked .Utterance 2: Oh , yeah . Had that dream .
Emotion: neutral

Utterance 1: Do not you realise what you are ... you are doing to yourself ?
Utterance 2: Hey , you know , I have had it with you guys and your cancer and your emphysema
and your heart disease .
Emotion: anger

Utterance 1: Oh , hey , do not do that ! Cut it out !
Utterance 2: It is worse than the thumb !
Emotion: disgust

Utterance 1: I am not moving out .
Utterance 2: You would tell me if you were moving out right
Emotion: fear

Utterance 1: So , what do you think ?
Utterance 2: I think It is the most beautiful table I have ever seen .
Emotion: joy

Utterance 1: No , wait , oh , what are we sorry about ?
Utterance 2: I do not know ... right , he is the pig !
Emotion: sadness

Utterance 1: No , wait , oh , what are we sorry about ?
Utterance 2: How did I not see this ?
Emotion: surprise


Utterance 1: UTT_1
Utterance 2: UTT_2
Emotion:
```

Figure 7: The prompt used to perform emotion classification with GPT-3.5 in the few-shot setting.

SCaLAR at SemEval-2024 Task 8: Unmasking the machine : Exploring the power of RoBERTa Ensemble for Detecting Machine Generated Text

Anand Kumar M
Department of IT
NITK, India

Abhin B
Artificial Intelligence
Department of IT
NITK, India

Sidhaarth Sredharan Murali
Artificial Intelligence
Department of IT
NITK, India

Abstract

SemEval SubtaskB, a shared task that is concerned with the detection of text generated by one out of the 5 different models - davinci, bloomz, chatGPT, cohere and dolly. This is an important task considering the boom of generative models in the current day scenario and their ability to draft mails, formal documents, write and qualify exams and many more which keep evolving every passing day. The purpose of classifying text as generated by which pretrained model helps in analyzing how each of the training data has affected the ability of the model in performing a certain given task. In the proposed approach, data augmentation was done in order to handle lengthier sentences and also labelling them with the same parent label. Upon the augmented data three RoBERTa models were trained on different segments of data which were then ensembled using a voting classifier based on their R2 score to achieve a higher accuracy than the individual models itself. The proposed model achieved an overall validation accuracy of 97.05% and testing accuracy of 76.25%.

1 Introduction

In the current day scenario, AI has noticed a major boom due the emergence of Large Language Models (LLMs) in the field of Natural Language Processing (NLP). These LLMs are capable of generating text with any given context that they have been trained on making them versatile to a lot of applications. LLMs have also showcased their unrivaled ability to code basic to complex programs. Many Large Language Models (LLMs) depend heavily on the data used for their training. Consequently, they may occasionally provide inaccurate information, especially in contexts where precision is crucial, such as sensitive or professional advice. Hence AI-generated text classification has become increasingly important due to the surge in the use of language models for content creation. Accurately identifying the source of a text, whether human-written or generated by a specific language model, is crucial for various applications, such as combating misinformation and plagiarism detection. Subtask B - Multi-Way Machine-Generated Text Classification shared task aims to not only detect text generated by these language models, but also specifically distinguish between outputs generated by different models. This in a real life scenario helps in determining the transparency and vulnerability of a model to attacks and reasoning as to why particular models perform in certain ways. Different contributions of the paper is as follows :-

- **Data Augmentation:** Data augmentation is a crucial task of increasing the volume of available data with specific manipulations which also helps build a more robust model able to tackle edge case scenarios. We propose a novel approach to handle long texts. We initially set a specific threshold to split them into smaller segments while preserving label information, ensuring efficient model training.

- **Ensemble Learning:** Ensemble learning as name suggests weaker models are brought together to achieve a better model with enhanced performance. We employ a weighted ensemble voting classifier that combines the predictions of multiple models trained on diverse validation sets, leading to improved generalizability and robustness.

We observe how effective and relevant Language models are in tackling Natural Language Processing tasks such as the current shared task when compared to other neural network based LSTM or other sequence models. Our final submission had a test accuracy of 76.25% and our standing was 18th position in the leader-board.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1135–1139

2 Background

Our work improves model generalizability in large-scale tasks by utilising insights from (Wang et al., 2024) and building on recent studies. Previously methods proposed by authors in (Ma et al., 2023) collected 500 scientific articles from 10 domains including biology, chemistry, IT and others and used chatGPT to paraphrase texts for each article. The authors extracted certain features such as perplexity, semantic document and six others to use classifiers on these extracted features. The authors used three classifiers: XGBoost, random forest, and multi-layer perceptron, to train and test models for detecting human-generated and AI-generated texts, as well as human-generated and AI-rephrased texts. They performed a 5-fold cross-validation and evaluated the models using accuracy and F1-score which majorly motivated our approach. Another work (Mindner et al., 2023) used similar techniques to the previous one while using school topics as their dataset. In their work (Abburi et al., 2023) the authors use ensemble neural model that generates probabilities from different pre-trained LLMs which are used as features to a TML classifier following it. Author in (Huimin et al., 2018) presents his work on text classification ensemble learning method based on multi-angle perturbation heterogeneous base classifiers and validates the effectiveness of the algorithm through experiments. In a similar work (Mohammed and Kora, 2022) the authors propose a new meta-learning ensemble method that fuses baseline deep learning models using 2-tiers of meta-classifiers.

Furthermore, our method for comprehending model decision-making in short text classification—particularly in identifying AI-generated content—is influenced by methods from works on short text classification. Authors in their work (Tang et al., 2022) use a sliding window to align the sentences with the labels and preserve the edge characteristics of the long text. Another work in the same field (Shorten et al., 2021) categorizes text data augmentations into symbolic and neural methods. Symbolic methods use rules or discrete data structures to form new examples, while neural methods use auxiliary neural networks to sample new data. Our research aims to advance the development of robust and adaptable machine learning models customised to particular tasks through this synthesis of diverse viewpoints. These viewpoints are then combined back again with the help of ensembling ensuring no loss of data.

3 Methodology

Our methodology majorly focuses on exploiting Pre-trained language models such as the RoBERTa model (Liu et al., 2019) and enhancing its performance through a much simpler traditional approach of ensemble learning. We worked on the M4 based dataset with our methodology (Wang et al., 2023) The ensemble model shows better performance compared to all 3 RoBERTa-base models which were trained on different segments of augmented data. It reduces over-fitting and increases interpretability for any given task.

3.1 RoBERTaForSequenceClassification

RoBERTa which stands for Robustly Optimized BERT Approach is a variant of the famous BERT model (Devlin et al., 2018) which was developed by Google in 2018. RoBERTa was later introduced by researchers at Facebook AI in 2019. It builds upon the architecture of BERT while bringing in few major changes. It uses Dynamic masking strategies and removes the Next Sentence Prediction (NSP) in its pre-training step. It is further pre-trained with larger amounts of data with larger mini-batch size. The novelty of RoBERTa lies in its ability to achieve state-of-the-art performance on various natural language understanding benchmarks by leveraging advancements in pre-training techniques and model architectures. RoBERTa continues to employ similar tokenizing technique as BERT with WordPiece Encoder (WPE). RoBERTa as a base model in itself gives out embedding for a given sentence or a word as it is only composed of Encoder architecture.

RoBERTaForSequenceClassifcation consists of a classification head on top of the base RoBERTa model. This classification head maps the backbone outputs to logits suitable for a classification task based on the number of labels provided.

3.2 Ensemble Learning

Ensemble learning is a machine learning technique that combines multiple individual models to obtain a model with enhanced performance which is more robust as well. It involves training several individual base models which are often referred to as experts on similar data and producing an aggregation out of those models based on their individual performances. The benefits of ensembling

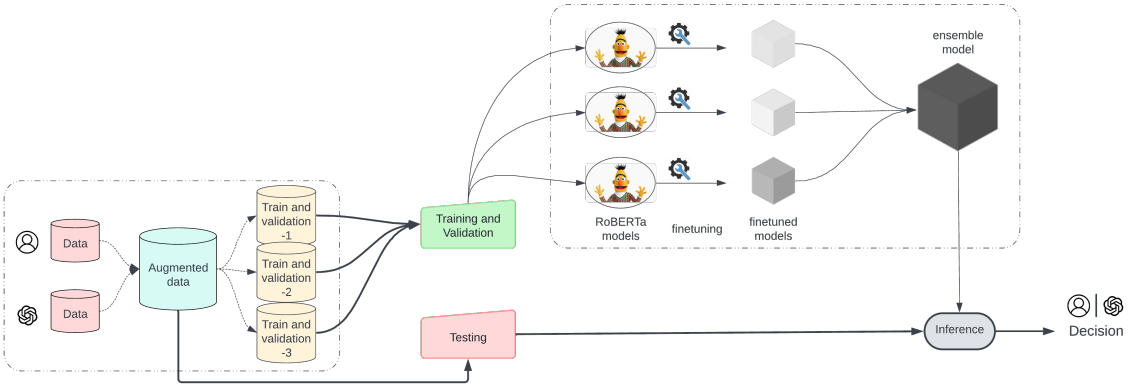


Figure 1: Proposed methodology for AI vs Human generated text Detection using weighted voting ensembling of RoBERTa classifiers

include improved generalization, more robustness compared to single models, and efficient as it compensates for the loss in performance of the poor learning algorithms. The common techniques in Ensemble learning including Bagging and Boosting. We specifically used the concept of Voting classifier which takes predictions from different models and has a specified weighting parameter based on which it gives out the final prediction. We implemented our own Voting classifier which scores the three RoBERTa models based on the R2-scores achieved by their predictions. R2-scores here are used as the weighting parameter for the prediction and thus we derive our final prediction out of the voting classifier.

4 System Overview

As mentioned, we first perform data augmentation. Before we get into the details of our experimental setup, we want to elaborate on different measures we took in order to augment our data. Data augmentation for training data was performed by carefully splitting the validation data while noting that there is no major imbalance in the class distribution. This was followed by training three different RoBERTa models on different combinations of training and validation dataset. We had 3 validation data splits namely *val1*, *val2* and *val3*. For *model1* we used *val2* and *val3* in training and *val1* for validating the *model1* and so on.

4.1 Data Augmentation

We implemented a data augmentation strategy to address instances in our dataset exceeding the token limit, ensuring no information loss while maintaining model compatibility. Given a dataset comprising 71,027 instances for training and 3,000 for validation, with some instances surpassing the 512-token limit, we devised a method to split these instances into k different segments. Utilizing the modulo operation, if an instance contains n tokens, $[n/512]$ determines the number of segments it will be divided into, while the remainder represents the number of tokens in the last augmented segment of the instance. This process yielded approximately 9985 additional instances for training and 188 for validation.

Subsequently, we merged the augmented training and validation sets to form a combined dataset of 81,012 training instances and 3188 validation instances. This validation dataset was then partitioned into three parts of which two-thirds are used for training alone with the complete training data and the remaining one-third is used for validation. Notably, each RoBERTa model was provided with a distinct subset of one-third of the validation data, thus adhering to a different k-fold validation scheme to enhance generalizability.

4.2 Implementation Details

The implementation of our method includes three vanilla RoBERTaForSequenceClassification models with 12 encoder layers with a classification head at the end were used. These models were trained on three different splits of two-thirds of validation data coupled with the training data. Each model was effectively trained on roughly 82000 samples with roughly 1060 validation instances. The voting classifier first takes in all three fine-tuned RoBERTa models and predicts on the complete validation set and analyzes the performance of each of the models based on their R2-score and constructs a weighted

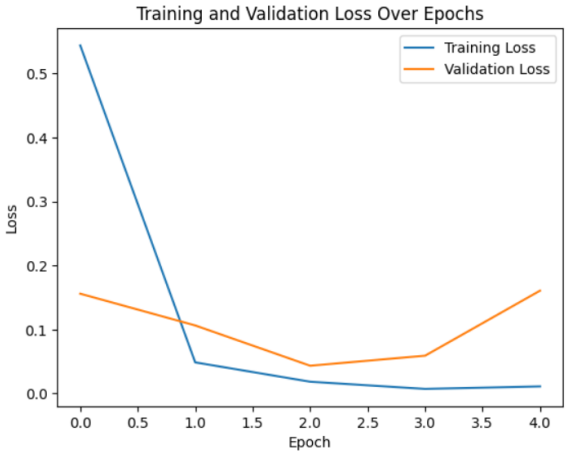


Figure 2: Training and validation loss observed over the RoBERTa model-1.

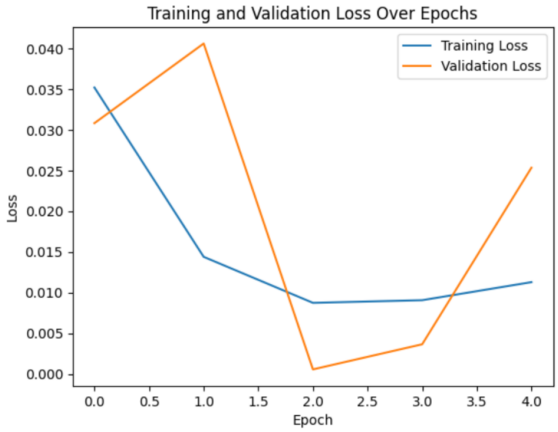


Figure 3: Training and validation loss observed over the RoBERTa model-2.

voting classifier which gives our final predictions. The R2 scores observed for each of the three models were 0.36, 0.29 and 0.35 which had roughly similar weight given to each of their predictions. The performance of each of the models were analyzed with the help of training and validation loss plots across training epochs.

5 Experimental Results

As a part of our experimental setup we used P100 GPU which is available through kaggle. Further we used the `RobertaForSequenceclassification` available through transformers library along with `RobertaTokenizerFast` for the modelling aspect. The learning rate used was a fixed one and we found it optimal at $1e-5$ along with $CrossEntropyLoss$. $AdamW$ optimizer was used with weight decay coefficient of $1e-2$ and $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Batch size of 20 was

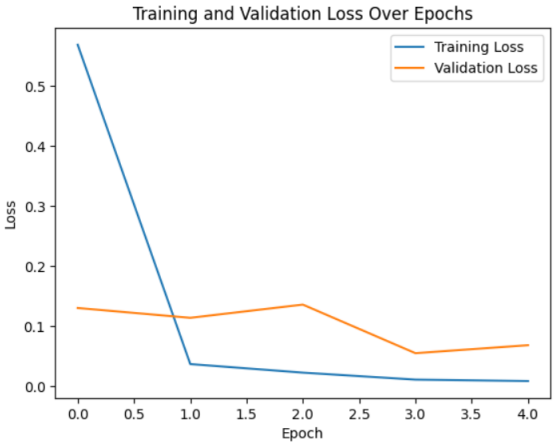


Figure 4: Training and validation loss observed over the RoBERTa model-3.

used for training and validation.

Our proposed methodology beat the baseline model which was a RoBERTa model with an average accuracy of 0.75. Our experimental results with respect to each of the RoBERTa model is displayed along with the improvement in performance with the use of R2-score based weighted voting classifier. In testing phase our model gave an accuracy of 76.25% which shows clear signs of over-fitting compared to 97.05% in validation accuracy.

Table 1: Proposed methodology performance comparison

Models	*Train Acc (%)*	*Val Acc (%)*
Baseline	75	75
RoBERTa-1	96.40	95.06
RoBERTa-2	93.64	92.10
RoBERTa-3	97.21	96.62
Voting Classifier (*proposed*)	**97.26**	**97.05**

6 Conclusion

In the proposed methodology, we beat the baseline RoBERTa model and further enhance the performance of the model using R2-score based Voting classifier. The model has performed well on the training data when compared to testing data which shows slight signs of over-fitting. In the light of ensemble learning for Pre-trained language models we see that the models are very sensitive to over-fitting hence should be used with caution. Techniques like early stopping and using data augmen-

tation. Further on embeddings from LLMs can be used to tackle this task more effectively.

References

Harika Abburi, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. 2023. Generative ai text classification using ensemble llm approaches. *arXiv preprint arXiv:2309.07755*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fan Huimin, Li Pengpeng, Zhao Yingze, and Li Danyang. 2018. An ensemble learning method for text classification based on heterogeneous classifiers. *International Journal of Advanced Network, Monitoring and Controls*, 3(1):130–134.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. 2023. Ai vs. human–differentiation analysis of scientific content generation. *arXiv*, 2301.

Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. 2023. Classification of human-and ai-generated texts: Investigating features for chatgpt. In *International Conference on Artificial Intelligence in Education Technology*, pages 152–170. Springer.

Ammar Mohammed and Rania Kora. 2022. An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*, 34(10):8825–8837.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8:1–34.

Changhao Tang, Kun Ma, Benkuan Cui, Ke Ji, and Ajith Abraham. 2022. Long text feature extraction network with data augmentation. *Applied Intelligence*, 52(15):17652–17667.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

PetKaz at SemEval-2024 Task 8: Can Linguistics Capture the Specifics of LLM-generated Text?

Kseniia Petukhova, Roman Kazakov, Ekaterina Kochmar
Mohamed bin Zayed University of Artificial Intelligence
{kseniia.petukhova, roman.kazakov, ekaterina.kochmar}@mbzuai.ac.ae

Abstract

In this paper, we present our submission to the SemEval-2024 Task 8 "Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection", focusing on the detection of machine-generated texts (MGTs) in English. Specifically, our approach relies on combining embeddings from the RoBERTa-base with diversity features and uses a resampled training set. We score 12th from 124 in the ranking for Subtask A (monolingual track), and our results show that our approach is generalizable across unseen models and domains, achieving an accuracy of 0.91. Our code is available at https://github.com/sachertort/petkaz-semeval-m4.

1 Introduction

SemEval-2024 Task 8 "Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection" (Wang et al., 2024) has focused on the detection of machine-generated texts (MGTs). In recent years, large language models (LLMs) have achieved human-level performance across multiple tasks, showing impressive capabilities in natural language understanding and generation (Minaee et al., 2024), including their abilities to generate high-quality content in such areas as news, social media, question-answering forums, educational, and even academic contexts. Often, text generated by LLMs is almost indistinguishable from that written by humans, especially along such dimensions as text fluency (Mitchell et al., 2023). Therefore, methods of automated MGT detection, intending to mitigate potential misuse of LLMs, are quickly gaining popularity. Automated MGT detection methods can be roughly split into black-box and white-box types, with the former being restricted to API-level access to LLMs and reliant on features extracted from machine-generated and human-written text samples for classification model training, and the latter focusing on zero-shot AI text detection without any additional training (see Section 2).

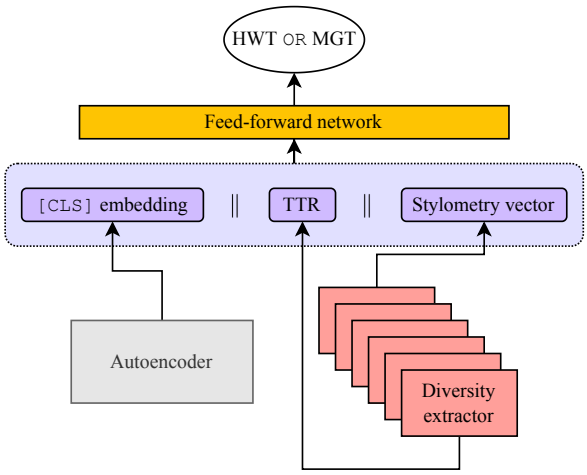


Figure 1: For each text, we get a [CLS] token embedding from an autoencoder model and extract vectors of linguistic features (e.g., lexical diversity, stylometry, etc.). Then, we pass the concatenated vector to a feed-forward network, whose output layer performs binary classification – HWT vs. MGT. The configurations of embeddings/features may vary between experiments.

For our submission to SemEval-2024 Task 8, the monolingual track of Subtask A, which focuses on MGT detection in English across a variety of domains and generative models, we have developed a system that can be categorized as a black-box detector and is based on a combination of embeddings, measures of lexical diversity, and careful selection of the training data (see Figure 1). We also present and discuss an extended set of linguistic features, including discourse and stylistic features, that we have experimented with during the development phase of the competition. The main motivation for using such a feature-based approach is that it helps us to focus on the fundamental differences between MGTs and human-written texts (HWTs) rather than capture the specifics of particular models.

Our results suggest that our best model, which uses diversity features and embeddings, outper-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1140–1147

forms a very competitive baseline introduced in this task (Wang et al., 2024), yielding an accuracy of 0.95 on the development and 0.91 on the test set. It brought us 12th place out of 124 teams participating in the shared task. Furthermore, our investigation shows that a model using no embeddings but relying on such linguistic features as entity grid and stylometry yields results that are on par with the baseline model.

The main contributions of our work are as follows: **(1)** we investigate the impact on the detection task of a variety of linguistically motivated features, ranging from widely used stylometric features to novel ones, including those based on high-level discourse analysis; and **(2)** we show how training data can be selected in an informative way to help models better distinguish between MGTs and HWTs.

2 Related Work

A comprehensive survey by Yang et al. (2023) categorizes detection methods into training-based classifiers, zero-shot detectors, and watermarking techniques, covering both black-box and white-box detection scenarios. This survey discusses a range of strategies, including mixed training, proxy models, and semantic embeddings, indicating ongoing challenges in scalability and robustness. Given the fast development of LLMs and their capabilities, of particular interest are innovations in zero-shot detection methods highlighted by Mitchell et al. (2023) and Su et al. (2023). In addition, Mitchell et al. (2023) present DetectGPT, utilizing perturbation discrepancies to discern MGTs, while Su et al. (2023) propose DetectLLM-LRR and DetectLLM-NPR, which advance zero-shot detection by harnessing log rank information.

Another relevant line of research investigates the use of linguistic and stylometric features, such as the ones overviewed in Bergsma et al. (2012), for MGT detection. For instance, Wang et al. (2023) explore the use of logistic regression with GLTR features (analyzing the distribution of token probabilities and their relative frequencies within specific probability ranges from a language model's output), stylistic characteristics, and NELA news verification features (style, complexity, bias, affect, morality, and event specifics) on the M4 dataset, and Liu et al. (2022) introduce a model exploiting text coherence, named entities and relation-aware graph convolutional networks under a low-resource setting for MGT detection.

3 Methodology

Our general pipeline, visualized in Figure 1, consists of the following components: (1) an autoencoder model fine-tuned on HWT vs. MGT classification task; (2) linguistic features extraction pipeline; and (3) embeddings and features combination passed through a feed-forward neural network. Below we describe some of these components in more detail.

3.1 Embeddings

We employ an autoencoder model. First, we fine-tune it on the HWT vs. MGT classification task, and then we use its `[CLS]` tokens' embeddings in a feed-forward model.

3.2 Features

We study the impact on the classification accuracy of several types of linguistically motivated features extracted from texts, including those based on: 1) text statistics; 2) readability; 3) stylometry; 4) lexical diversity; 5) rhetorical structure theory (RST); and 6) entity grid. Below we provide a description of the features and their relevance to the task.

Text statistics We compute the following:[1] 1) the number of difficult words (words that have more than two syllables and are not in the list of easy words[2] from Dale and Chall, 1948); 2) raw lexicon count (unique words in text); 3) raw sentence count. In Appendix A.1, we provide the values for HWTs and across models.

Readability We assess the readability of MGTs and HWTs guided by the hypothesis that HWTs are easier to read than MGTs. We calculate a range of common readability scores for both types of texts to assess their readability, including 1) Flesch Reading Ease Test (Flesch, 1979); 2) Flesch-Kincaid Grade Level Test (Kincaid et al., 1975); and 3) Linsear Write Metric (O'Hayre, 1966).

Stylometry For stylometric features, we use the approach proposed in Bergsma et al. (2012). Specifically, we collect all unigrams and bigrams from the texts and keep punctuation, stopwords, and Latin abbreviations (e.g., *i.e.*) unchanged. Then, we build two types of representations where

[1] We use Python's `textstat` library: https://pypi.org/project/textstat/.

[2] https://github.com/textstat/textstat/blob/main/textstat/resources/en/easy_words.txt

other words are replaced by their PoS tags and "spelling signatures" (forms of words; e.g., `xXX-dd` for *iOS-17*).[3] Then, log token frequencies (TFs) are computed for each text and passed to the maximum absolute scaler, and these sparse representations are used as features. For further processing, sparse matrices with stylometry features are reduced by truncated singular value decomposition to a dimensionality of 768. See Appendix A.2 for the analysis of stylometry features importance.

Lexical diversity Lexical diversity tells us how "rich" texts are in terms of vocabulary, i.e., whether they use rare words, or include a wide range of synonyms, epithets, terms, etc. There are a few measures widely used to measure lexical diversity, mostly based on the variants of the type-token ratio (TTR). We extract 10 features, such as TTR, Maas TTR, Hypergeometric distribution d (HDD; McCarthy and Jarvis, 2007), etc.[4] For an in-depth overview, see McCarthy and Jarvis's (2010) study on lexical diversity assessment.

RST features In rhetorical structure theory (RST), proposed in Mann and Thompson (1988), texts are analyzed in terms of hierarchical structures, which represent the organization of information and text flow. These structures are made up of elementary discourse units (EDUs) connected through rhetorical relations, which include "elaboration", "contrast", "cause", "result", etc. Using an open-source sentence-level RST parser (Lin et al., 2019), we count the occurrences of various relations in each text and divide them by the total number of sentences in the text.

Entity grid Finally, we use the entity grid algorithm to analyze the coherence of text by capturing patterns of entity distribution (Barzilay and Lapata, 2005). This method transforms a text into sequences of entity transitions, documenting the distribution, syntax, and reference information of discourse entities. Entities from texts are first tagged with their syntactic roles[5] and categorized into three types: subject (`s`), object (`o`), and other (`x`). The next step involves examining the transition of entities' roles across consecutive sentence pairs. This includes transitions like subject-to-object, object-to-other, subject-to-none, among others. Finally, we calculate the frequency of each transition type for all entities by dividing the total count of each transition type by the number of sentence pairs.

3.3 Feed-forward neural network

Finally, we use a concatenation of embeddings and vectors representing combinations of various features described above and pass them as input to a feed-forward neural network. Then, the output layer performs binary classification.

4 Data

Shared task organizers have used an extension of the M4 dataset (Wang et al., 2023),[6] which covers a range of domains (including *WikiHow*, *Wikipedia*, *Reddit*, *arXiv*, *PeerRead*, and *Outfox*) and texts generated by a number of LLMs (including `ChatGPT`, `Cohere`, `Davinci003`, `Dolly-v2`, `BLOOMZ`, and `GPT-4`) as well as written by humans. Overall, the training set is roughly balanced between HWTs and MGTs, with 53% being HWTs and with the number of HWTs being around 5 times higher than that of texts generated by any single LLM for each of the domains. The only exception is *PeerRead*, where the distribution of texts generated by each LLM and written by humans is about the same. At the same time, the distribution is exactly 50%:50% for HWTs:MGTs in the development set, and 47.5%:52.5% for HWTs:MGTs in the test set. In addition, while both training and development sets cover a range of domains, the test set is limited to *Outfox* only.

A curious case of WikiHow Before running the experiments, we further investigate how the training data is composed. According to Wang et al. (2023), LLMs were provided with relatively short inputs to generate texts across various domains: for example, with titles for *Wikipedia* articles and *arXiv* papers, titles and abstracts for *PeerRead* articles, etc. On the one hand, we observe a high level of parallelism in the training data across HWTs and texts generated by various models, and on the other, we note that there is little consistency in what models generate in certain domains: for example, provided with a name of a personality they generate quite different *Wikipedia* entries, which do

[3] The pre-processing was done using `spaCy`: https://spacy.io.

[4] Using Python's `lexical_diversity` library: https://github.com/kristopherkyle/lexical_diversity.

[5] Noun coreference is resolved using `spaCy` (https://spacy.io) and `neuralcoref` (https://spacy.io/universe/project/neuralcoref).

[6] https://github.com/mbzuai-nlp/M4

not only differ from the correspondent HWTs but also vary from one LLM to another (see examples in Appendix B, Table 5). In contrast, texts in the *WikiHow* domain appear to be more similar to each other across LLMs, which can be explained either by the way the data was generated (using titles and headlines as prompts to produce MGTs) or by the fact that there are fewer ways to explain *How to do X?* compared to the tasks in other domains. Moreover, our experiments with in-domain training of the MGT detection classifier suggest that the best results can be obtained when it is trained on the *WikiHow* domain. We follow up on these observations and create a customized training subset by using all MGTs from the original data and limiting HWTs to the texts from the *WikiHow* domain only. This results in a training set of 56,406 MGTs and 15,499 HWTs, with the distribution between each LLM and humans being roughly 1:1.

5 Experiments

5.1 Experimental setup

As the source of embeddings, we use `roberta-base`[7] (Liu et al., 2019) fine-tuned within the baseline framework[8] over 3 epochs with the learning rate of $2e$-5 and L_2 norm of the weights being 0.01. The feed-forward neural network with two hidden layers accompanied by a ReLU activation function is then trained with the learning rate $5e$-5, L_2 norm of the weights 0.01, and early stopping after 25 epochs. Each hidden layer has batch normalization and a dropout of 0.5. We use `PyTorch`[9] (Paszke et al., 2019) for all training and evaluation steps.

Following up on our observations on the *WikiHow* subset described in Section 4, we conduct two series of experiments and train the feed-forward network on: 1) the *full* training set; and 2) the *reduced* training set where we use MGTs from all domains and HWTs from *WikiHow* only.

5.2 Experiments on the development set

The evaluation results of our model with different feature configurations applied to the development set are presented in Table 1. Several observations are due at this point.

[7] https://huggingface.co/FacebookAI/roberta-base

[8] https://github.com/mbzuai-nlp/SemEval2024-task8/tree/main/subtaskA/baseline

[9] https://pytorch.org

Configuration	Full train	Reduced train
`feat`	0.60	0.60
`sty`	0.68	0.57
`sty` ‖ `feat`	0.69	0.60
`sty` ‖ `div`	0.65	0.72
`sty` ‖ `read`	0.67	0.61
`sty` ‖ `rst`	0.64	0.57
`sty` ‖ `ent`	0.73	0.56
`emb`	0.74	0.83
`emb` ‖ `sty`	0.73	0.82
`emb` ‖ `feat`	0.76	0.90
`emb` ‖ `div`	0.73	**0.95**
`emb` ‖ `read`	0.72	0.81
`emb` ‖ `rst`	0.73	0.81
`emb` ‖ `ent`	0.73	0.82
Baseline	0.74	–

Table 1: Accuracy of different configurations and the baseline on the development set. `feat` stands for all features except stylometry, `sty` – stylometry, `div` – lexical diversity, `read` – text statistics and readability, `rst` – RST, `ent` – entity grid, `emb` – embeddings (see Section 3.2).

First of all, we note that the highest accuracy of 0.95 is achieved with the model trained on the *reduced* training set using a combination of embeddings and diversity features. This does not mean that lexical diversity is necessarily the most powerful among linguistic features, but it suggests that it complements embedding representations better than other linguistic features. Moreover, it is the only feature type that increases the accuracy obtained with embeddings only. Finally, we also note that with the linguistic features, our model can outperform a competitive baseline used by the task organizers, which sets the accuracy at 0.74.

Secondly, stylometry features turn out to be the best linguistic feature type when used on their own: the accuracy with `sty` is 0.68 vs. 0.6 with `feat`. These representations reflect some general patterns of word types used in texts. However, it seems like they alone are not enough for effective classification, at least when applied to texts generated by modern LLMs. Notably, the configuration that combines stylometry with entity grid features (`sty + ent`) demonstrates performance that is nearly identical to the baseline employing a pre-trained language model (0.73 vs. 0.74), suggesting that entity grid adds further information about text coherence. Other features like RST do not seem to help distinguish MGTs from HWTs. This finding

Configuration	Train	Accuracy	F_1
`emb` ‖ `div`	reduced	**0.91**	**0.92**
`sty` ‖ `ent`	full	0.84	0.85
Baseline	full	0.88	–

Table 2: Metrics on the test set. The first row is our main submitted configuration. The organizers do not report only the baseline's F_1 score.

suggests that the frequency or efficacy with which humans and models employ rhetorical structures is comparable.

Finally, we observe that the performance of the model using `emb` features always increases if it is trained on the *reduced* set. This determines the model configuration for our final submission.

6 Results

Table 2 presents accuracy on the test set obtained with two configurations: a model using embeddings and lexical diversity features trained on the reduced training set, and a model using stylometry and entity grid features trained on the full training set, which showed promising results on the development set. **The former one is our main configuration: our team has submitted its predictions for the test set and scored 12th in the shared task (out of 124 teams).** This model outperforms the organizers' baseline, which sets the accuracy at 0.88. However, we note that the latter model, which relies on linguistic features only and does not employ any pre-trained language model, also shows promising results, further strengthening our hypothesis that linguistic features are able to capture important properties of LLM-generated texts.

6.1 Analysis

We further analyze the performance of our best model across different LLMs on the test set, as illustrated in Figure 2. The results show that our model accurately identifies texts from `Dolly-v2`, `Cohere`, and `ChatGPT` as machine-generated, and achieves near-perfect classification precision on texts from `GPT-4` and `Davinci003`. `BLOOMZ` is the only model that presents a problem for our classifier, with an 8% misclassification rate. Additionally, we observe that 18% of HWTs are incorrectly classified as being generated by machines. This shows the remarkable generalizability of our approach compared to Wang et al. (2023), who reported that "it is

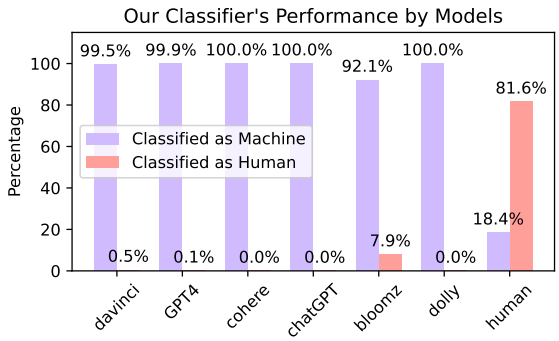


Figure 2: Performance of our classifier across models.

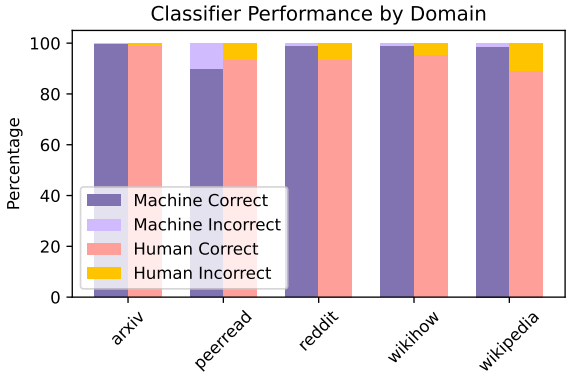


Figure 3: Performance of our classifier across domains (on the development set).

challenging for detectors to generalize well on unseen examples if they are either from different domains or are generated by different large language models. In such cases, detectors tend to misclassify machine-generated text as human-written".

Furthermore, we evaluate our model's performance across domains (Figure 3). Our analysis reveals that we can accurately identify all MGTs and nearly perfectly recognize HWTs from *arXiv*. Our classifiers face the biggest difficulties when classifying MGTs from *PeerRead* and HWTs from *Wikipedia*. These results are aligned with those reported in Wang et al. (2023), who also found that training on *Wikipedia* leads to the worst out-of-domain accuracy.

In summary, our classifier demonstrates generalizability, performing well on both previously unseen models (`GPT-4` and `BLOOMZ`) and domains (with all texts in the test set being from *Outfox*).

7 Conclusions

When developing the models for our submission to the SemEval-2024 Task 8, we have primarily focused on: (1) the contribution of linguistic features to the task, and (2) the selection of the informative training data. Our results suggest that models using

only linguistic features (specifically, those based on stylometry and entity grid) can perform competitively on this task, while careful selection of the training data helps improve the performance of the models that rely on embeddings. This shared task demonstrates that it is possible to distinguish between HWTs and MGTs, but the results also suggest promising avenues for future research, including in-depth analysis of the training data selection techniques and expansion of the linguistic features.

Limitations

Our work is limited to the English language only as we opted to participate in a single Subtask of SemEval-2024 Task 8. In addition, this work is only limited to the domains and LLMs included in the shared task data, therefore, the generalizability of our approach beyond these domains and LLMs will need to be verified in future experiments.

Acknowledgements

We would like to express our gratitude to Ted Briscoe for inspiring us with the idea that linguistics could be of help and for engaging in discussions with us. We are grateful to Mohamed bin Zayed University of Artificial Intelligence (MBZUAI) for supporting this work. We also thank the anonymous reviewers for their valuable feedback.

References

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 141–148, Ann Arbor, Michigan. Association for Computational Linguistics.

Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric analysis of scientific articles. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–337, Montréal, Canada. Association for Computational Linguistics.

Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28.

Rudolf Franz Flesch. 1979. *How to write plain English: a book for lawyers and consumers*. University of Canterbury.

Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Xiang Lin, Shafiq Joty, Prathyusha Jwalapuram, and M Saiful Bari. 2019. A unified linear-time framework for sentence-level discourse parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4200, Florence, Italy. Association for Computational Linguistics.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Yu Lan, and Chao Shen. 2022. CoCo: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Philip McCarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42:381–92.

Philip M. McCarthy and Scott Jarvis. 2007. vocd: A theoretical and empirical evaluation. *Language Testing*, 24(4):459–488.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature.

John O'Hayre. 1966. *Gobbledygook Has Gotta Go*. U.S. Department of the Interior, Bureau of Land Management.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. *ArXiv preprint*, abs/2306.05540.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *ArXiv preprint*, abs/2305.14902.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023. A survey on detection of LLMs-generated content. *ArXiv preprint*, abs/2310.15654.

A Features Analysis

A.1 Text statistics across models

Table 3 shows various text statistics calculated on the training set. It can be seen that HWTs have higher values than all MGTs across all these metrics.

Model	DW	LC	SC
ChatGPT	64	350	19
Cohere	37	256	13
Davinci003	58	315	16
Dolly-v2	54	342	18
Human	91	582	30

Table 3: Text statistics on the training set. DW = difficult words (mean), LC = lexicon count (mean), SC = sentence count (mean).

A.2 Stylometry features importance

Stylometry features are passed to a linear SVM classifier[10] to extract coefficients that may be interpreted as feature importances. Table 4 presents the most important features for MGTs and HWTs in the case of binary classification: for example, we can see that proper nouns are mostly associated with HWTs. It also makes it clear how the features are ordered by importance.

MGT feature	Wt.	HWT feature	Wt.
How to	3.28	NOUN SPACE	-4.12
SPACE *How*	2.34	SPACE	-4.12
NUM VERB	2.07	xxxx	-3.93
Xxxxx *the*	2.00	SPACE ADJ	-3.10
How	1.78	SPACE PROPN	-3.10
SPACE NUM	1.77	*the* SPACE	-2.67
Well	1.57	NOUN	-2.57
Xxx *the*	1.38	NUM SPACE	-2.21
dd Xxxxx	1.37	PROPN SPACE	-2.15
NOUN *you*	1.37	_XXX_d	-2.14

Table 4: Stylometric features highly weighted by the binary SVM classifier.

B Data Statistics

Table 5 shows some examples of parallel texts extracted from three domains represented in the training set (*WikiHow*, *Wikipedia*, and *PeerRead*). As explained in Wang et al. (2023), the data for each

[10]From scikit-learn (Pedregosa et al., 2011): https://scikit-learn.org.

	WikiHow
ChatGPT	Buying Virtual Console games for your Nintendo Wii is a fun and easy process that can net you some classic games to play on your console. [...]
Cohere	How to Buy Virtual Console Games for Nintendo Wii The Nintendo Wii has a feature called the Virtual Console that allows you to download and play games from past Nintendo consoles, such as the Nintendo Entertainment System. [...]
Davinci003	How to Buy Virtual Console Games for Nintendo Wii Most people know that Nintendo's library of classic titles is available on the Wii platform through the Virtual Console. [...]
Dolly-v2	Find a few Wii Points cards from game retailers like GameStop., Make sure your Wii is online and on a secure connection if possible. [...]
Human	They are about $20 a card. Or, if you want to just buy points with your credit card, Skip down to the section, With a Credit Card. [...]
	Wikipedia
ChatGPT	William Whitehouse was a 19th-century British engineer and inventor who made significant contributions to the field of hydraulics. [...]
Cohere	William Whitehouse (1567-1648) was an English scholar, schoolmaster, and Anglican clergyman. [...]
Davinci003	William Whitehouse (August 6, 1590 - May 18, 1676) was an English priest, scholar and biblical commentator. [...]
Dolly-v2	William Whitehouse (born William John Whitehouse; 15 July 1944) is an English musician, singer and songwriter. [...]
Human	William Edward Whitehouse (20 May 1859 - 12 January 1935) was an English cellist. [...]
	PeerRead
ChatGPT	The paper "End-to-End Learnable Histogram Filters" aims to introduce a novel approach that enables histogram filters to be learnable end-to-end. [...]
Cohere	This paper addresses the problem of designing end-to-end learnable histogram filters. [...]
Davinci003	This paper presents an interesting approach to combining problem-specific algorithms with machine learning techniques to find a balance between data efficiency and generality. [...]
Dolly-v2	The paper End-to-End Learnable Histogram Filters demonstrates an interesting technique for reducing photo noise without blurring the image. [...]
Human	We are retracting our paper "End-to-End Learnable Histogram Filters" from ICLR to submit a revised version to another venue. [...]

Table 5: Parallel HWTs and texts generated by different LLMs in the training set extracted from selected domains.

domain was generated in a different way (for instance, using an article title only in some cases, and more extended inputs in others). We observe that there is much less consistency between the outputs generated by different LLMs in such domains as *Wikipedia* and *PeerRead* than in *WikiHow*. For instance, in the case of generated *Wikipedia* articles, the models cannot even agree on what personality they are describing (which is obvious from the very first sentences of such generated articles), while in the case of generated reviews from *PeerRead*, article descriptions also exhibit high diversity in the way they are presented in the review. At the same time, we hypothesize that generating texts for the *WikiHow* domain, describing *How to do X?*, results in higher consistency in the models' outputs, which is exemplified in Table 5.

SLPL SHROOM at SemEval2024 Task 06: A comprehensive study on models ability to detect hallucination

Pouya Fallah[1], Soroush Gooran[1], Mohammad Jafarinasab[1], Pouya Sadeghi[2], Reza Farnia[1], Amirreza Tarabkhah[3], Zainab Sadat Taghavi[1], and Hossein Sameti[1]

[1]Sharif University of Technology
[2]University of Tehran
[3]Amirkabir University of Technology

Abstract

Language models, particularly generative models, are susceptible to hallucinations, generating outputs that contradict factual knowledge or the source text. This study explores methods for detecting hallucinations in three SemEval-2024 Task 6 tasks: Machine Translation, Definition Modeling, and Paraphrase Generation. We evaluate two methods: semantic similarity between the generated text and factual references, and an ensemble of language models that judge each other's outputs. Our results show that semantic similarity achieves moderate accuracy and correlation scores in trial data, while the ensemble method offers insights into the complexities of hallucination detection but falls short of expectations. This work highlights the challenges of hallucination detection and underscores the need for further research in this critical area.

1 Introduction

While Natural Language Generation (NLG) has empowered machines to craft increasingly sophisticated text, transforming the NLP landscape, a dark undercurrent lingers - the phenomenon of hallucinations. In NLG, hallucinations refer to fabricated or misleading content woven into a generated text, deviating sharply from reality (Laurer et al., 2023)(Varshney et al., 2023). These fictional elements, despite seeming plausible due to their learned patterns, threaten the very core of NLG's promise: **reliability** and **truthfulness** (Ji et al., 2023). Imagine summarizing news articles riddled with fictional details or translating medical instructions brimming with inaccuracies. Such scenarios underscore the profound and potentially dangerous implications of hallucinations within NLG, making their detection and mitigation an urgent priority (Huang et al., 2023).

The specter of hallucinations looms large over NLG, particularly in domains demanding unyielding accuracy and safety. Imagine a medical summary riddled with invented details or medication instructions marred by mistranslations – these scenarios, chillingly possible, could directly jeopardize patient well-being (Ji et al., 2023). Recognizing this critical threat, researchers have embarked on a mission to untangle the complexities of hallucinations, developing methods for their detection and ultimately, prevention (Huang et al., 2023).

This paper dives headfirst into the challenge of hallucination detection within NLG, leveraging recent advancements in the field. We employ diverse methodologies to unmask these fictional elements in SemEval 2024 Task 6 evaluation data. Firstly, we assess the semantic similarity between generated text (hypotheses) and the provided reference outputs, gauging their alignment in meaning. Secondly, we harness the power of cosine similarity of embeddings, allowing us to capture subtle semantic nuances and relationships within text representations. Furthermore, we integrate Natural Language Inference (NLI), analyzing whether the generated text logically implies or contradicts factual information. Additionally, we utilize Large Language Models (LLMs) to discern context similarity, leveraging their inherent language understanding to identify inconsistencies that might point toward hallucinations. But our approach goes beyond established techniques. We introduce a novel judgment LLM framework, where one LLM acts as a discerning judge, scrutinizing the outputs of other LLMs for signs of hallucination. This innovative approach leverages the collective strengths of multiple models while introducing an element of meta-reasoning to the detection process.

2 Related Work

Several approaches have been proposed for detecting hallucinations in NLG text, categorized into different access levels to the model:

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1148–1154

Knowledge-based Approaches: Fact verification compares generated text with information from a domain-specific knowledge base. This approach can be effective but requires substantial knowledge bases and may not generalize well to unseen domains.

Classification approach: (Liu et al., 2022) created a dataset specifically for hallucination detection, but it has not been very successful.

White-box and Grey-box Approaches: Hidden state analysis: (Azaria and Mitchell, 2023) use an MLP classifier on the LLM's hidden states to predict truthfulness. This requires access to internal model states and may not apply to all architectures.

Token probabilities: Grey-box methods analyze the token probabilities generated by the LLM, assuming factual sentences contain high probability tokens. However, this can be unreliable for complex or ambiguous phrases.

Black-box Approaches: Self-evaluation: (Kadavath et al., 2022) propose asking the LLM itself to assess the likelihood of its output being correct. While promising, this method relies on the LLM's self-awareness and may not be reliable for all models.

Proxy model: This approach uses a publicly available LLM to estimate the token probabilities of the black-box model's output and infer its factual consistency. However, its accuracy depends on the proxy model's similarity to the black-box model.

Selfcheckgpt (Manakul et al., 2023) introduced a black box approach. The main idea of this study is that if the LLM is trained on a concept if multiple responses are taken from it, the samples will be similar and include consistent facts. Whereas if it is hallucinating, the samples will be different and contradictory. Therefore, several samples are taken from the LLM, and by measuring the information consistency between the responses, we can understand whether they are factual or hallucinated.

Our Approach: This paper builds upon existing work by combining elements from different categories. We leverage information consistency within multiple LLM responses, inspired by Selfcheckgpt (Manakul et al., 2023), but introduce a novel "judgment LLM" framework that goes beyond self-evaluation by employing one LLM to scrutinize the outputs of others. This approach aims to address previous methods' limitations by leveraging multiple models' collective strengths and introducing meta-reasoning into the detection process.

3 Task Description

The SHROOM challenge shines a light on a crucial hurdle in natural language generation (NLG) - pinpointing seemingly correct text that holds inaccurate meaning, often referred to as "misleading outputs." We, alongside other participants, are tasked with detecting these "semantic hallucinations," even when they are flawlessly written and grammatically sound.

The challenge focuses on "fluent overgeneration," where generated text, despite being linguistically coherent, strays from the intended semantic meaning. Participants operate in a "post hoc" setting, assuming models have already been trained and their outputs generated.

This is where we step in – to identify these misleading texts amidst seemingly accurate ones. This is critical to ensure the truthfulness and reliability of NLG outputs, especially in real-world applications.

The SHROOM challenge presents a two-pronged approach to tackle fluent overgeneration hallucinations: model-aware and model-agnostic tracks. Participants can choose to leverage knowledge of the model or not, depending on their approach. This multifaceted assessment covers three key NLG domains: definition modeling, machine translation, and paraphrase generation. The challenge provides a rich dataset including:

- **Checkpoints:** Model snapshots at different training stages
- **Inputs:** Prompts or texts used for generation
- **References:** Human-written outputs that represent the intended meaning
- **Outputs:** The actual text generated by various models trained with varying accuracy

Furthermore, a dedicated development set with binary annotations by multiple annotators ensures robust evaluation. This collaborative effort results in a majority vote gold label, boosting the dataset's credibility. Ultimately, the SHROOM challenge strives to develop effective solutions for combating semantic hallucinations generated by Large Language Models.

4 Proposed Systems

In our study, we tried to do the hallucination detection task in two separate methods and tried to compare and analyze their results:

4.1 Semantic Similarity method

Detecting hallucinations based on semantic similarity involves evaluating the coherence between language model outputs and reference data. In our study, we utilized this approach due to the availability of reference outputs. By assessing the semantic alignment between the generated text and the reference data, we aimed to discern instances of hallucination where the model output diverged from the intended meaning.

LaBSE The Language-Agnostic BERT Sentence Embedding (LaBSE) model is a dual-encoder approach based on pre-trained transformers, further refined for machine translation ranking. LaBSE excels at encoding sentences into fixed-length vectors while capturing semantic information across various languages (Feng et al., 2020). We employed LaBSE, particularly due to one of our tasks being machine translation (MT). By calculating the cosine similarity between model outputs and reference data, we determined the hallucination score in our study.

LLMs We utilized Zephyr-7B-β (Tunstall et al., 2023) and Mistral-7B (Jiang et al., 2023a) language models (LLMs) to assess the semantic similarity between model outputs and reference data, assigning a score between 0 and 1.

4.2 Natural Language Inference (NLI)

Due to the insufficient data available for hallucination detection, one proposed approach is to utilize models trained for similar tasks. **Natural Language Inference (NLI)** is one such task. In NLI, a language model assesses the relationship between text fragments, namely the premise and the hypothesis. This task involves multiclass classification aimed at determining whether the hypothesis can be inferred from, contradicts, or remains neutral to the premise.

The concept here involves treating reference data as the premise and model outputs as the hypothesis, then utilizing the probability of one of the outputs as a score to determine hallucination. We employed a **DeBERTa-v3** model, fine-tuned on datasets like MNLI, FERVER, ANLI, WANLI, and LingNLI (Laurer et al., 2023), to calculate the entailment score, which serves as the inverse of the hallucination score.

Note that employing NLI models in hallucination detection is not a novel concept and has been utilized by researchers in recent years (Ji et al., 2023). Here, we employed it to compare with our proposed judgment method.

4.3 Ensemble LLMs: The Judgment Method

To improve LLMs' reasoning and decision-making abilities, we explored two approaches: intrinsic self-correction and multi-agent feedback. We acknowledge that existing LLMs struggle with self-correction, and due to our LLMs' similarities, we believe they might mislead each other in a multi-agent setup. Inspired by the (Jiang et al., 2023b)article, we designed experiments using ensemble models. We asked LLMs to generate results multiple times with confidence scores, and finally extracted the best result.

We used two "commentator" models to assess the consistency between two sentences in detail. Based on their answers (yes/no/maybe, score, and explanation), a "judge" model (Mistral 7B or Zephyr 7B) performed hallucination detection.

While Mistral 7B and Llama2 (Touvron et al., 2023) provided three responses per data point, Zephyr only gave one. The advantage of multiple responses is the potential for higher accuracy. In cases of agreement, we considered the model confident and non-hallucinating. Contradictions suggested hallucination, but instead of simply discarding opinions, we devised rules to combine the responses.

We implemented the judgment method with three configurations:

Composition 1: Mistral and Zephyr commented, with Mistral judging based on their comments (label, score, description output).

Composition 2: Same as 1, but Zephyr judged.

Composition 3: Llama2 and Mistral commented, with Zephyr choosing whose opinion was more reliable.

5 Experiments and Experimental Setup

5.1 Semantic Similarity

In semantic similarity methods, we consider the target output in MT and DM tasks, and the input in PG tasks as reference. The similarity

score between the reference and hypothesis for each data point is computed, and 1 minus this score is considered as the probability of hallucination. Probabilities below 0.5 are classified as "Not Hallucination", while others are labeled as "Hallucination".

Using the **LaBSE** model, we obtained embeddings of the reference and hypothesis, and the cosine similarity of these two embeddings was considered as the similarity score.

Using the prompt "Is the Sentence supported by the Context above?" we asked each **LLM** (Zephyr and Mistral) to provide a score between 1 and 5 determining the similarity of the reference and hypothesis. These values were then normalized to range between 0 and 1.

5.2 Natural Language Inference (NLI)

For the NLI method, we used the reference as the premise and the hypothesis of each record as the hypothesis in the NLI model. This model outputs three probabilities which determine the probability of entailment, neutral, and contradiction between the two inputs. We utilized the probability of entailment for hallucination detection as it yielded better results compared to the other two options on the validation data.

5.3 Judgement method

Commenting LLMs were prompted to check whether the sentence was supported by the context and were asked to return a label, a score between 1 and 5, and an explanation.

Prompt for commenting LLMs:

Answer the following question using this JSON format: answer: (yes, no or maybe), score: (an integer number between 1 and 5, which 1 is for not supported and 5 is for fully supported), description: (a description for your answer). question Is the Sentence supported by the Context above?

Three outputs were taken from Llama2 and Mistral for each data, and these three outputs were converted into one with rules.
In the first two compositions, Mistral and Zephyr were commenters and the judge was asked to return an output with the same label, score, and explanation format after reviewing their explanations.
To see the result of self-correction, the judge was selected from among the commentators. The first time was Mistral and the second time was Zephyr.
This was the prompt:

Two experts are asked whether the given sentence supports the given context or not. We received two responses from these two experts. According to the explanations of these two experts, what is your decision? return your response in this JSON format label: (yes/no), score: (an integer number between 1 and 5, which 1 is for not supported and 5 is for full supported), explanation: (text).

In the third combination, the commentators were Llama2 and Mistral, and Zephyr was the judge. This time, we changed the prompt to the judge so that he chose only one of the two opinions as the more correct opinion.
This was the judge's prompt:

Answer the following question.
question
I asked two experts to determine whether the Sentence is supported by the Context or not.
Above are their explanations.Now judge which one gave a better reason. Give me just the index of the best expert with no explanations using this JSON format: index: (an integer number between 0 and 1, which 0 is for the first, 1 is for the second).

6 Results and Analysis

Semantic Similarity and NLI:
Table 1 and Figure 1 showcases the results of hallucination classification on trial data for each method we employed. Based on these findings, the Semantic Similarity method, utilizing models like LaBSE and Zephyr, demonstrates moderate accuracy and correlation scores. While LaBSE holds promise due to its renowned semantic similarity capabilities, there's room for improvement.
Notably, among the two Language Learning Models (LLMs) utilized in the semantic similarity approach, Zephyr yielded considerably better results than Mistral. This was also evident in the validation data, which influenced our decision to incorporate it into all our judgmental method experiments.

The DeBERTa model or NLI method outperforms all other methods, suggesting that incorporating natural language inference strengthens our ability

	Model Aware		Model Agnostic	
	Accuracy	Correlation(ρ)	Accuracy	Correlation(ρ)
LaBSE	0.706	0.426	0.658	0.464
Zephyr	0.700	0.370	0.694	0.423
Mistral	0.630	0.213	0.568	0.183
DeBERTa-v3(NLI Model)	**0.777**	**0.661**	**0.780**	**0.689**
Mistral Judge (Zephyr & Mistral Reasons)	0.644	0.291	0.610	0.250
Zephyr Judge (Zephyr & Mistral Reasons)	0.686	0.352	0.692	0.405
Zephyr Judge (LlaMa2 & Mistral Reasons)	0.624	0.293	0.548	0.249

Table 1: Experiment Results

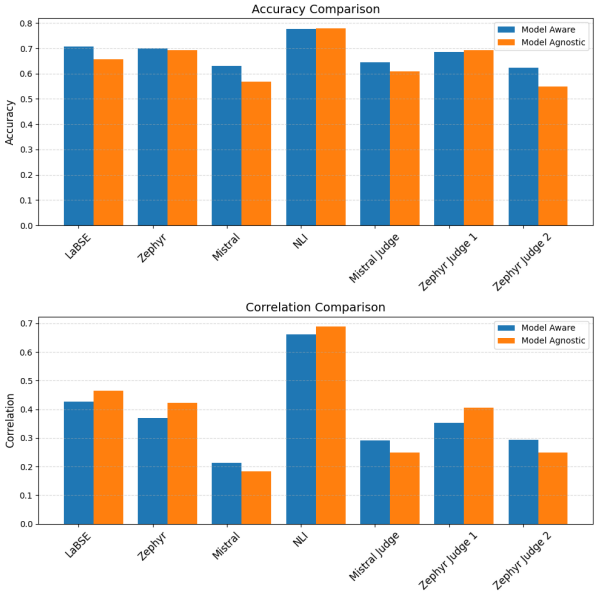


Figure 1: Comparison of accuracy and correlation scores across multiple models in model-aware and model-agnostic datasets.

to discern hallucinations by capturing semantic relationships between generated text and reference data.

Ensemble LLMs and the Judgment Method:

Furthermore, our study explored the effectiveness of Ensemble LLMs utilizing a Judgment Method. Surprisingly, the results indicate that Ensemble LLMs' performance wasn't superior to the previous method; in fact, it was even lower. Zephyr, acting as a judge, exhibited lower accuracy and correlation scores compared to Zephyr alone. While Mistral, in the role of a judge, showed improved performance compared to Mistral alone, it still falls short of methods like DeBERTa and the LaBSE model, suggesting limitations in this approach's effectiveness for hallucination detection.

7 Conclusion

This study investigated three distinct methods for hallucination detection in language models: the Semantic Similarity method, NLI and the Ensemble LLMs with Judgment method. By analyzing and comparing these approaches, we gained valuable insights into their efficacy and suitability for identifying hallucinatory content in model-generated text.

Semantic Similarity Method:
The utilization of pre-trained models such as LaBSE or large language models (LLMs) like Zephyr has demonstrated the potential for hallucination detection by assessing coherence between generated text and reference data through the Semantic Similarity method. Our findings underscore the effectiveness of employing specialized embedding models like LaBSE, which consists of approximately 500 million parameters, yielding comparable results to LLMs like Zephyr with 7 billion parameters. This highlights the efficiency of utilizing specialized embedding models for such tasks. However, while the semantic similarity method has shown moderate

success, it falls short of being deemed the optimal choice for hallucination detection. Relying solely on similarity may not adequately capture all forms of hallucination and could prove insufficient across various tasks and scenarios. It's worth noting that exploring these limitations is beyond the scope of this paper and warrants further investigation by other researchers.

NLI:
In conclusion, our findings underscore the efficacy of the NLI method as the optimal model for our study, indicating its potential utility in hallucination detection through entailment scoring. However, similar to the Semantic Similarity method, it is essential to acknowledge the inherent limitations in extrapolating the concept of entailment to the domain of hallucination detection. While NLI datasets offer valuable insights, they may not encompass the full complexity of hallucination phenomena. Therefore, while NLI tasks present promising avenues for further exploration in this area, additional research is warranted to ascertain their applicability and effectiveness in comprehensive hallucination detection frameworks.

Ensemble LLMs with Judgment Method:
This novel approach introduced multi-agent feedback and ensemble modeling for hallucination detection. LLMs acted as commentators, providing input to a "judge" model for final decision-making, aiming to enhance individual models' reasoning and decision-making. While not exceeding initial expectations, our experiments yielded valuable insights into the ensemble's effectiveness, with varying accuracy and correlation depending on composition and judging strategies.

Discussion and Future Directions:
Although the performance of the Ensemble LLMs with Judgment method wasn't as promising as envisioned, it sheds light on the complexities of hallucination detection and the limitations of current methods. One of the key challenges in these methods is finding the optimal prompt to detect hallucinations in the language model, and the utilization of prompt engineering methods can be beneficial in this regard. The potential for improved results using larger, more capable LLMs suggests avenues for future exploration.

Overall, this study contributes to addressing challenges posed by hallucinations in language models. By evaluating and comparing distinct detection methodologies, we highlight the strengths and weaknesses of each approach, paving the way for future research and development in this crucial area.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2023. Building efficient universal classifiers with natural language inference. *arXiv preprint arXiv:2312.17543*.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

INGEOTEC at SemEval-2024 Task 1: Bag of Words and Transformers

Daniela Moctezuma[†] and **Eric S. Tellez**[‡] and **Mario Graff**[‡]
[†] CentroGEO, Aguascalientes, México
[‡] CONACyT - INFOTEC, Aguascalientes, México
dmoctezuma@centrogeo.edu.mx
{eric.tellez,mario.graff}@infotec.mx

Abstract

Understanding the meaning of a written message is crucial in solving problems related to Natural Language Processing; the relatedness of two or more messages is a semantic problem tackled with supervised and unsupervised learning. This paper outlines our submissions to the Semantic Textual Relatedness (STR) challenge at SemEval 2024, which is devoted to evaluating the degree of semantic similarity and relatedness between two sentences across multiple languages. We use two main strategies in our submissions. The first approach is based on the Bag-of-Word scheme, while the second one uses pre-trained Transformers for text representation. We found some attractive results, especially in cases where different models adjust better to certain languages over others.

1 Introduction

Semantics refers to the meaning of language, including words, phrases, sentences, and overall text. Understanding semantics is essential for text comprehension and communication, as it allows us to interpret the intended meaning of a message accurately. Semantic relatedness measures how similar the meaning of two words or phrases is. It is based on the idea that words related in meaning tend to co-occur frequently in language or even have some causal relation connecting them. For example, *cat* and *dog* are semantically related because they refer to common household pets. Measuring semantic relatedness is essential for many natural language processing tasks, such as information retrieval, question answering, and machine translation.

Relatedness models play a crucial role in natural language processing (NLP). These models determine the degree of similarity or relatedness between two pieces of text. One of the critical benefits of relatedness models is that they can help improve the performance of NLP applications by providing more relevant and accurate results. For example, relatedness models can be used in information retrieval to rank search results based on their relevance to the user's query. Similarly, relatedness models can help identify the most relevant answer to a user's question while solving the question-answering problem.

A method based on corpus-based word similarity and string similarity, as well as their order, is proposed in (Islam and Inkpen, 2008). For string similarity, the authors used the longest common subsequence (LCS) in three ways to weight, i.e., the work is based on measuring the shared order of words. The word mover's distance, see (Kusner et al., 2015), reformulates the problem of comparing two sequences of words to an optimal transportation problem. It represents two sentences with its word embeddings and computes its optimal alignment using a dynamic programming solution; while it is pretty promising, it does not require sentences to be of some fixed size and works with a myriad of possible word embeddings. However, the technique was revisited by (Sato et al., 2022) and found diverse issues that limit its effectiveness.

Kenter and De Rijke (Kenter and de Rijke, 2015) have used word embeddings (word2vec) and external sources of semantic knowledge to represent text messages and meta-features. They aim to interpret proximity in the generated latent space as semantic similarity.

More recently, the Transformer deep neural networks have become a powerful alternative to both lexical and semantic approaches; the approach is based on a stack of encoders and decoders layers and the self-attention procedure (Vaswani et al., 2017). Transformers have a high computational cost, primarily for training. The first Transformer

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1155–1159
June 20-21, 2024 ©2024 Association for Computational Linguistics

that can be pre-trained and fine-tuned to match different tasks is BERT (Devlin et al., 2018a); after BERT, the high cost of training is paid once since fine-tuning needs less computational power and much less data. The interested reader should review the BERT manuscript and the seminal paper about pre-training NLP models (Howard and Ruder, 2018).

Fine-tuning BERT for classification or regression tasks is straightforward, not because it is a simple architecture but due to the myriad of literature, repositories, and examples showing how to do it [1]. However, its usage for sentence similarity needs to produce a vector that works fine for the task, and it is not trivial to produce one with its standard matrix output. The sentence transformers (Reimers and Gurevych, 2019) use siamese networks to create effective sentence vector embeddings for tasks working with pairs of sentences, for instance, similarity search and clustering.

In their research, Chandrasekaran and Mago (Chandrasekaran and Mago, 2021) have surveyed the evolution of semantic similarity methods, reviewing various NLP approaches, including traditional techniques and those found in machine learning and deep learning. They have provided a detailed study describing the strengths and weaknesses of each approach.

A binary version of the relatedness tasks is as follows: given a pair of sentences u and v, predicting true if u and v are related and false otherwise. A more elaborated task is to predict a relatedness score $\mathsf{rel}(u, v) \in [0..1]$, where values near zero mean for no relation and values near 1 mean for total relatedness. The latter definition is used in the Semantic Text Relatedness Task 1 (Ousidhoum et al., 2024b) at SemEval-2024, which asked for predicting relatedness scores for nine multilingual datasets; in particular, we tackled the problem as a supervised learning problem, i.e., we focused only on subtask 1 using the data for the nine languages(for more details about dataset see (Ousidhoum et al., 2024a)).

This document outlines the strategies we employed for the Semantic Textual Relatedness (STR) challenge in SemEval 2024, specifically the track A for the nine languages considered. To tackle this task, we utilized two distinct approaches: a transformer method for the English and Spanish languages, and an EvoMSA (Graff et al., 2020) solution for the remaining languages, which include Algerian Arabic, Amharic, Hausa, Kinyarwanda, Marathi, Moroccan Arabic, and Telugu.

The paper is organized as follows: Section 2 describes all our solutions to task 1. Section 3 shows our experimental results. Finally, Section 4 concludes our results and findings.

2 System overview

Nowadays, one of the most common approaches to dealing with natural language processing (NLP) problems is those Transformer-based language models. However, the pre-training procedure of this kind of language model needs a vast text corpus, and therefore, it may be impossible now to train them properly in many languages. In these cases, models based on counting and computing statistics may be more robust. We used Transformers for languages we know have large language models explicitly created for that language; for other datasets, we use a back-propagation optimized EvoMSA model for each one.

2.1 Out transformer-based approach

Our model was trained as a regression using the following procedure. For each pair, we extracted the sentence embedding for each sentence and evaluated the cosine similarity between pairs of embeddings. We trained a linear Support Vector Machine regressor using the cosine similarity to learn and predict the given relatedness score.

We tested several Transformer models but chose those that gave us the best performance, all of them were used directly as Hugging Face indicated. In this case, the best ones were microsoft-mpnet-base(Song et al., 2020) and multilingual BERT (Devlin et al., 2018b).

The microsoft-mpnet-base (MPNet) is a pre-training model, it tries to deal with the dependency on the predicted tokens and takes auxiliary position info into account to see a full sentence and reduce the position difference (Song et al., 2020).

The multilingual BERT is a well-known transformers model pre-trained on a large corpus of multilingual data self-supervised. In overview, it has two main tasks, MLM (Masked Language Modeling) and NSP (Next Sentence Prediction) (Devlin et al., 2018b), nevertheless, we just used the embedding representation to deal with the competition's task.

[1] For instance, one of the main sources of pre-trained Transformer models and documentation about them is the Hugging face project `huggingface.co`

2.2 Our EvoMSA approach

We use our EvoMSA framework for languages different than English and Spanish. EvoMSA models can be tailored for the dataset or pre-trained. Our pre-trained models were constructed using a small tweet corpus per language collected from the public Twitter stream. In addition, our EvoMSA models can be lexical based on bag-of-words (BoW) or semantic based on creating embeddings using numerous pre-trained classifiers in several self-supervised problems. Our BoW model produces highly sparse vectors where each component represents a token in the vocabulary. At the same time, our semantic representation (Dense) produces dense vectors created with the decision function of several binary classifiers, each one learned in a set of self-supervised tasks. The precise construction of EvoMSA models is detailed in (Graff et al., 2023).

Our approach to tackle the relatedness problem is to state it as a regression problem combining BoW and Dense representations using the following expression:

$$V = \left(S^\top \cdot S_Q, T^\top \cdot T_Q, (D \odot D_Q) \cdot \theta_1\right) \quad (1)$$

$$\hat{V} = \sigma\left(\frac{V}{\|V\|}\theta_2 + \beta\right) \quad (2)$$

where S, S_Q, T, and T_Q are sparse BoW matrices encoding pairs of sentences with statistics from pre-trained vocabularies (S) and training set-based vocabularies (T); D and D_Q are Dense matrices corresponding to pair of sentences, again computed with models pre-trained. Matrices without sub-indices mean for the first sentence in the pair, and those matrices with sub-indices Q mean for the second pair's element. The trainable parameters θ_1 and θ_2 are vectors, and β is a trainable scalar. Also, σ is the sigmoid function. We use differential programming with the JAX framework (Bradbury et al., 2018) for the Python programming language to train our models using $1 - \mathsf{pearson\_correlation}(\cdot, \cdot)$ as a loss function. In particular, we initialize θ_2 and β as the optimized parameters of a Linear Support Vector Machine parameters (solved firstly per each model with these parameters) and then θ_1 as a vector of ones instead of the typical random initialization to help on fine-tuning parameters. We call this model as One+.

We performed multiple modifications to this scheme and also found that defining V as $(T \odot T_Q) \cdot \theta_1$ results in a very competitive option. This model is called One-B. Note that this approach works only with the training set and does not require any pre-trained models.

3 Experimental results

We considered our two approaches with several expression variants for our EvoMSA-based approach and several models for our Transformer-based approach. Our model selection finds the best models using $1 - \mathsf{pearson\_correlation}$ with k-folds cross-validation along multilingual datasets. We selected the One+ and One-B model expressions since they demonstrated to be robust among many others coming from One+, also note that One-B works only with the training set.

In particular, the Transformer approach was better for Spanish and English datasets. We tested with several BERT, SBERT, and MPNet models before selecting *microsoft/mpnet-base* model for English and the multilingual BERT model, specifically the *bert-base-multilingual-cased*.[2]

Table 1 lists our best approaches for the different languages for the relatedness tasks in the third column. We can observe how transformers work fine for English and Spanish, languages with plenty of available models and data. For the rest of the languages, our EvoMSA approach performs better, but we can also observe that the simpler model One-B performs better in several datasets; this may be because of the lack of pre-trained models for that language, in particular, for languages with low available resources.

Table 1 also reports the Spearman correlation score and the global rank under the *dev* and *eval* datasets. Here, we can observe how our approach achieves different language ranks. In particular, we reached among the top ten results for Algerian and Moroccan Arabic. The English model is not among the top, but the score is not very different from the best ones. Note how One-B is competitive for Amharic, Hausa, Kinyarwanda, and Telugu, working without additional data.

It is important to say that, we did not achieve outstanding results, so, further analysis cannot be done, we saw lower results in those languages less studied, and more generalized models performed better in most common languages such as English and Spanish. Also, in the case of less-known languages, a simpler strategy was the best such as the Bag-of-Words-based proposed approach.

[2]Available on huggingface and its *Transformers* library.

Code	Language	Model	Spearman	Rank Dev Correlation (dev/eval)	Rank Eval
Arq	Algerian Arabic	One+	0.574 / 0.566	8	10
Amh	Amharic	One-B	0.676 / 0.702	19	15
Eng	English	Transformer	0.789 / 0.809	35	29
Hau	Hausa	One-B	0.547 / 0.576	20	15
Kin	Kinyarwanda	One-B	0.430 / 0.630	14	12
Mar	Marathi	One+	0.750 / 0.784	21	20
Ary	Moroccan Arabic	One+	0.820 / 0.811	12	9
Spa	Spanish	Transformer	0.701 / 0.678	7	13
Tel	Telugu	One-B	0.818 / 0.801	10	14

Table 1: Best model and results for each language dataset for the relatedness prediction problem.

4 Conclusion

This manuscript describes our participation in Task 1 of Semantic Textual Relatedness (STR) at SemEval 2024. We used two main approaches: a transformer-based approach and an EvoMSA-based one. The latter has lexical and semantic representations, with variants using pre-training and fully learned from the training data. Our transformer solution works better for Spanish and English, while our EvoMSA works better for the other languages. In particular, we support low-resource languages using our EvoMSA without pre-trained models. Our competitive results give evidence suggesting that languages with fewer resources can benefit from models that do not require an enormous corpus to be trained; this can be an alternative to large models. Nevertheless, this is a very complex task, and better efforts could be made in the future.

References

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs.

Dhivya Chandrasekaran and Vijay Mago. 2021. Evolution of semantic similarity—a survey. *ACM Comput. Surv.*, 54(2).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova Google, and A I Language. 2018a. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Technical report.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Mario Graff, Sabino Miranda-Jiménez, Eric S. Tellez, and Daniela Moctezuma. 2020. EvoMSA: A Multilingual Evolutionary Approach for Sentiment Analysis. *Computational Intelligence Magazine*, 15(1):76 – 88.

Mario Graff, Daniela Moctezuma, Eric Tellez, and Sabino Miranda. 2023. Ingeotec at DA-VINCIS: Bag-of-Words Classifiers. *CEUR Workshop Proceeding*, 3496:1–10.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *Preprint*, arXiv:1801.06146.

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data*, 2(2).

Tom Kenter and Maarten de Rijke. 2015. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 1411–1420, New York, NY, USA. Association for Computing Machinery.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *Preprint*, arXiv:2402.08638.

Nedjma Ousidhoum, Mohamed Abdalla Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. 2022. Re-evaluating word mover's distance. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19231–19249. PMLR.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

OctavianB at SemEval-2024 Task 6: An exploration of humanlike qualities in hallucinated LLM texts

Octavian Brodoceanu
University of Bucharest Faculty of Mathematics and Informatics
octavian.brodoceanu@gmail.com

Abstract

The objective of the SHROOM shared task (Mickus et al., 2024) is to identify sequences of text generated by Large Language Models that contain incorrect, nonfactual, or fabricated information. These sequences, referred to as 'hallucinations', are characterized by lower probabilities assigned to the outputs, as demonstrated by research (Varshney et al., 2023). This discrepancy highlights a possible contrast in the language used between hallucinated and non-hallucinated texts. The aim of this paper is to investigate whether hallucinated responses exhibit phrasing and patterns that more closely resemble those of machine-generated text rather than coherent, human-like language.

1 Introduction

The SHROOM shared task, as described by Mickus et al. (2024), has as its objective 'detecting grammatically sound output that contains incorrect semantic information (i.e. unsupported or inconsistent with the source input), with or without having access to the model that produced the output'. This type of output is encompassed by generations often referred to as "hallucinations". According to Varshney et al.,in the context of language models, hallucinations refer to the generation of text or responses that seem syntactically sound, fluent, and natural but are factually incorrect, nonsensical, or unfaithful to the provided source input. (Maynez et al., 2020; Holtzman et al., 2023; Koehn and Knowles, 2017) With the advent of mainstream Large Language Models brought upon by OpenAIs ChatGPT (Brown et al., 2020), it is a relatively new and important topic in this context. Apart from the possible spread of misinformation, being able to make this distinction is crucial for the adoption of Large Language Models in domains highly sensitive to misinformation, such as the medical, jurisdictional or financial fields. Possible repercussions include medical misdiagnosis, fictitious financial or legal advice, or exploitation by bad actors in order to deceive users.

In our approach to solving this task, we employ two methods. The first method involves utilizing models trained for detecting machine-generated text in order to distinguish between regular and hallucinated sequences. The other involves using looking at the loss of the hypothesis as scored by an LLM(Large Language Model), in the hope that generations with low probabilities can be properly tagged as hallucinations. This method was introduced by Fu et al. (2020) as GPTScore as a way to get a numerical assessment of an aspect in a given text.

When it comes to the first method, the hypothesis to be tested is that patterns which help differentiate machine generated texts will be transferable to the task at hand. The rationale is as follows: the training data of the model is human-written text, therefore deviations from the training set could be detected in this manner. From the experimental results on the model aware track, the performance of this method yielded a score of 0.483. This is below the baseline achieved using an LLM to label the generations. These results could stem from the hypothesis itself, or the fact that the model is not able to differentiate the texts of newer LLMs.

The second method was employed after the end of the competition, as a way to further explore the dataset and its characteristics. It was based on the success of Ji et al., who used a similar approach in a reprompting system meant to reduce hallucinations. On the validation data, it yielded an accuracy of 0.686 without the target reference and 0.702 when it was included in the prompt on the model aware track.

2 Background and Dataset

The objective of the SHROOM shared task is to detect hallucinations in two distinct datasets: one that is model agnostic and one that is model aware.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1160–1165

Both of the datasets consist of text pertaining to 3 tasks: DM - 'Definition Modeling' - which involves providing the definition of a word given surrounding context, PG - 'Paraphrase Generation' - in which the generated text is meant to be a paraphrase of the input, and MT - 'Machine Translation' - in which the task is to translate a given sequence. The text provided for the definition modeling and the paraphrase generation task types is in English. For the machine translation task, the prompt is provided in the native language and the hypothesis and target are both in English. The model-aware validation dataset consists of 499 datapoints, while the model-agnostic version has 501 datapoints. The test sets both have 1500 samples.

Train Model Aware		
Column Name	**Description**	**Data Type**
hyp	Generated sequence	String
tgt	Desired target sequence	String
src	Prompt sequence	String
ref	Target column	Categorical
task	Prompt task type	Categorical
model	LLM model name	Categorical

Table 1: Description of Model Aware Dataset Columns

The main difference between the datasets is that the 'model' column is not present in the model agnostic version. This distinction is not relevant to the experiments presented in this paper, as no data is used apart from the 'hyp' - generated sequence column.

The datasets are split by the organizers in train, validation and test sets respectively, with the validation and test sets containing human-annotated labels. The probability of hallucination is defined as the average of the label given by each annotator, and the final label is chosen by majority vote from said labels. The validation and test set have 5 such labels per entry.

An example datapoint consists of the input 'Resembling or characteristic of a weasel.' - corresponding to this input, the output is structured as per Table 2.

Output		
label	**p(Hallucination)**	**id**
Not Hallucination	0.470	1

Table 2: Example of Model Aware Dataset Row

3 Related Work

Due to the importance and the relative novelty of the LLM hallucination detection task, there are many recently proposed ways to alleviate the issue. Ji et al. proposed a system for preventing hallucinations via self reflection, by using GPTScore as a way to gauge aspects such as factuality and consistency. Using a community sourced body of knowledge, for example wikipedia, in order to greatly enhance context (Semnani et al., 2023). Perturbations to the input to check for model self consistency (Zhang et al., 2023). Segmenting the generations and reprompting to check for consistency also appears to have lead to good results. (Wei et al., 2023; Zhou et al., 2023; Khot et al., 2023) Looking at the log probabilities of the output words to detect low-confidence generations (Varshney et al., 2023) has also been proposed, an approach very similar to one of the two methods used.

4 Methodology

4.1 Method 1: Generated Text Detection

The first method involves using a pretrained model for distinguishing machine-generated text. The decision to use this type of model stemmed in part from the similarity of the two tasks. Considering the fact that the training set for Large Language Models is often entirely human-written, deviations from the dataset - which are a possible cause of hallucinations - should appear as machine-like generations.

The model used during inference is 'roberta-large-openai-detector' (Solaiman et al., 2019). It is a a RoBERTa-large (Liu et al., 2019) model that has been trained in order to differentiate between texts generated by the Large Language Model GPT2 (Radford et al., 2019) after its inception. As the authors explain, it is able to distinguish texts generated by the LLM with 95% accuracy. The use of this model is, however, a limitation of the experiment. As cited by the authors (Solaiman et al., 2019) 'The model developers also report finding that classifying content from larger models is more difficult, suggesting that detection with automated tools like this model will be increasingly difficult as model sizes increase.' It should also be noted

that due to the nature of the MT - Machine Translation task, hallucinations of this type are unlikely to be picked up by the model.

Input is taken as the 'hyp' hypothesis column in the dataset. Since it is under the form of simple text, it will be tokenized using the 'roberta-large-openai-detector' tokenizer with padding and truncation. No other changes were made to the text.

Outputs are represented by the logits resulting from passing the tokenized input sequences through the model. The logits are then passed through a softmax function in order to obtain probabilities attributed to each class (0 - not generated/not hallucinated, 1 - generated/hallucinated). The class with the highest probability is saved as the 'label' and the probability of the input belonging to the 'hallucinated' label is 'p(hallucination)'. In the case of the test set, the id of the sequence is added to the structure to be added to the json.

4.2 Method 2: GPTScore

The second method involves prompting a pre-trained LLM with a task and checking the loss attributed to the predefined output.

The prompt is comprised of: instruction, demos, input and output.

Instruction prompts were constructed for each of the 3 tasks in the dataset, for example: "The following is a Definition Modeling task. Please focus on capturing the correct meaning based on the surrounding context in the original text. "

For each of the 3 tasks, demos were constructed by randomly sampling 3 datapoints from a subset of the validation dataset. This subset involves rows labeled "Not hallucination" by all five human annotators, in the case of the positive examples, and "Hallucination" in the case of the negative examples.

The input is the prompt sequence provided in the dataset. The output is the response provided in the same datapoint.

As an example, a prompt with no demos would be: Give the definition for the specified words in the given context. The answer for "The sides of the casket were covered with heavy black broadcloth , with velvet caps , presenting a deep contrast to the rich surmountings . What is the meaning of surmounting ?" is "A sloping top ."

The resulting output of the method is defined as the average of the logprobs of the output sequence (i.e. "A sloping top"). Naturally, the output score is predicated on the model doing the evaluation, with more accurate models having a higher chance of giving better results.

Optionally, the target sequence can be added to the prompt. Although this increases performance, as we would expect, it changes the use of the method to that of evaluation.

The models used include a quantized version of **Mistral-7B** and **OpenHermes-13B**. After generating the scores for each of the inputs, the goal is to employ simple binary classification. Logistic regression and SVM were tested, with logistic regression consistently giving superior results.

5 Experimental Setup

The first method is fully unsupervised, and therefore does not require calibration on the training set. The second method requires us to have a subset of labeled data to determine the score threshold.

For evaluating our models, we used the metrics proposed by the organizers: accuracy, based on the labels and Spearman's Rho, based on the probabilities assigned to each entry.

6 Results

The outputs on the test set model-aware track yielded a score of 0.483. The results on the validation dataset were the following:

Validation Set Results		
Track	**Accuracy**	**Rho**
Model Agnostic	0.545	0.033
Model Aware	0.465	-0.145

Table 3: Valdiation dataset results

The results for method 2 are shown in Tables 4 and 5:

Model Aware Track Validation Set Results				
Model	**Total**	**PG**	**MT**	**DM**
Mistral-7B	0.686	**0.776**	0.696	0.638
OpenHermes-13B	0.688	**0.808**	0.691	0.643

Table 4: Validation model aware dataset accuracy results

Model Aware Track Validation Set Results				
Model	**Total**	**PG**	**MT**	**DM**
Mistral-7B	0.687	0.704	**0.812**	0.657
OpenHermes-13B	0.701	0.688	**0.802**	0.625

Table 5: Validation model agnostic dataset accuracy results

Valdiation task aware results			
Track	**DM**	**PG**	**MT**
Model Agnostic	0.540	**0.624**	0.497
Model Aware	0.489	**0.608**	0.345

Table 6: Valdiation dataset task-aware accuracy results

6.1 Track results analysis

As evident, the model agnostic accuracy surpasses that of the model-aware track by a considerable margin. This difference could be due to statistical noise, as both results seem to be within 5% of the expected value for random binary attribution i.e. 50%. One competing hypothesis would be that the hidden distribution of the agnostic track allows for the model to better differentiate between the two classes.

Inferred sample label distributions in the form of 'Hallucinated'/'Not Hallucinated' are the following: 177/322 for model agnostic and 240/261 for model aware. For comparison, the real distributions are 218/281 model-agnostic track and 206/295. The fact that the model-aware results exhibit a near 50-50 split, in contrast to the model-agnostic track, whose distribution is closer to that of the real set, leads some credence to the hypothesis that the inference model is be able to detect relevant patterns.

Spearman correlation is calculated using the 'p(hallucination)' column. In the context of the proposed model, this is the probability assigned to class 1 ('Hallucination'). From the resulting values, it is evident that the probabilities of the reference and input display little to no correlation, with both values being near 0. From this, we come to the conclusion that the proposed method is not suitable for inferring the probability of hallucination.

6.2 Task-aware results

In order to investigate if the task had any impact on the performance of the model, standard accuracy was calculated for each separate subset of sequences. This was done on both the model aware and model agnostic track. The results are as per Table 6:

6.3 Task-aware results analysis

DM - 'Definition Modeling' showcases better performance on the agnostic dataset. As stated above, this could be attributed to random noise, or a difference in the distribution of the dataset.

PG - 'Paraphrase Generation' displayed the highest accuracy out of all three tasks on both the model-agnostic and model-aware tracks. It is a consistent and large enough improvement from the random baseline to be considered significant.

MT - 'Machine Translation' task results were the lowest, with the model reaching the expected random outcome of 50% on the model agnostic track. The results for the Model Aware Track showed an unexpected and significant difference, 15.5% from the random baseline. Low performance is to be expected, as this task requires the least free generation. These results could be due to the fact that the LLM is simply translating sequences that have been written by humans, and this requires less 'creative' generation on its part.

One plausible explanation for why the method has displayed superior performance on the PG task could be attributed to its inherently free-form nature compared to the other tasks. Definition Modeling is an information retrieval adjacent task, and Machine Translation leaves little room for interpretation, apart from cases of ambiguous wording. This suggests that the proposed method may have potential applications in specific tasks given to LLMs.

As per the results showcased in 4 and 5, we notice the increased performance when using OpenHermes-13B. This is to be expected, as it is the larger model, and the efficacy of the method is predicated on the quality of the certainty attributed by each model. We notice a leap in accuracy for certain tasks, Paraphrase Generation in the case of the model aware track, and Machine Translation in the case of the model agnostic track. This may once again be due to a difference in the distributions of the two tracks.

7 Additional experiments

Post competition, in addition to method 2, we have attempted to further finetune the RoBERTa model

to see if we can improve performance. In order to do this, we have used the dataset provided by Liang et al.. It is comprised of 749 datapoints, containing text generated by GPT3 and GPT4, as well as human written text. Before any additional operations, the model has an accuracy of 0.539 on this set, further confirming the fact that more powerful models and methods of detecting machine generated text are needed for use on newer LLMs. The dataset was split in a 9-1 ratio of training-validation data. After finetuning for 3 epochs the accuracy on the evaluation dataset reached an accuracy of 0.591. From this we can conclude the model is not able to properly learn. The results of the finetuning are shown in Table 7.

Validation Set Results	
Track	**Accuracy**
Model Agnostic	0.436
Model Aware	0.411

Table 7: Validation Set Results

The reason for why the model does not seem to learn can either be due to the model itself, or the generations are too humanlike to be told apart. As for the results of the finetuning, the performance of the model has dropped significantly on both datasets.

8 Conclusions

We have proposed two methods, the first based on using models pretrained on generated text detection, and the second based on looking at the confidence displayed by the LLM under the form of logits. Reviewing the results, we can assess that the tasks of generated text detection and hallucination detection showcase too large of a divergence for the approaches to generally be used interchangeably. However, the results on the Paraphrase Generation task may warrant further investigation into the use of models pretrained for text generation detection for the hallucination detection task. Concerning the second method we have explored, it has showcased promising results on specific tasks in each track, which may warrant use in an ensemble method.

8.1 Limitations

The main limitation of this experiment was the model used for inference. As it was trained to discriminate the generations of GPT2, which is a significantly smaller model compared to the current Language Models.

8.2 Future Work

In future work, we might explore model finetuning on newer datasets used for discerning between human and machine-generated texts, as well as finetuning pretrained models on labeled hallucination related tasks.

Another simple improvement would be utilizing pretrained models able to better distinguish between generated and human written texts.

We may test the first method on other free-form generation tasks, as this seems to be a strong suit.

We may also look into newer methods for detecting machine generated text, that account for the leaps made by the recent advancements in LLMs.

We may further investigate the discrepancy in accuracy for the provided datasets when using GPTScore.

Acknowledgements

I would like to thank Ana Sabina Uban for kickstarting the idea for this work.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv:2005.14165v4*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2020. Gptscore: Evaluate as you desire.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2023. The curious case of neural text degeneration.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating llm hallucination via self reflection.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zouv. 2023. Gpt detectors are biased against non-native english writers. *arXiv:2304.02819*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, , and Ryan McDonaldr. 2020. On faithfulness and factuality in abstractive summarization.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sina Semnani, Violet Yao, Heidi Zhang, and Monica Lam. 2023. Wikichat: Stopping the hallucination of large language model chatbots by few-shot grounding on wikipedia.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *arXiv:1908.09203v2*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, , and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley A. Malin, and Sricharan Kumar. 2023. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chii. 2023. Least-to-most prompting enables complex reasoning in large language models.

FI Group at SemEval-2024 Task 8: A Syntactically Motivated Architecture for Multilingual Machine-Generated Text Detection

Maha Ben-Fares[1,2], Urchade Zaratiana[2,3], Simon D. Hernandez [2] and Pierre Holat[2,3]

ETIS, CY Cergy Paris Université - Pontoise, France[1]; FI Group, Puteaux, France[2]
LIPN, Université Sorbonne Paris Nord, Villetaneuse, France[3]
maha.ben-fares@cyu.fr, zaratiana@lipn.fr, {simon.hernandez, pierre.holat}@fi-group.com

Abstract

In this paper, we present the description of our proposed system for Subtask A - multilingual track at SemEval-2024 Task 8, which aims to classify if text has been generated by an AI or Human. Our approach treats binary text classification as token-level prediction, with the final classification being the average of token-level predictions. Through the use of rich representations of pre-trained transformers, our model is trained to selectively aggregate information from across different layers to score individual tokens, given that each layer may contain distinct information. Notably, our model demonstrates competitive performance on the test dataset, achieving an accuracy score of 95.8%. Furthermore, it secures the 2nd position in the multilingual track of Subtask A, with a mere 0.1% behind the leading system.

1 Introduction

The evolution and widespread adoption of Generative Pre-trained Transformers, notably with the release of ChatGPT have significantly influenced the landscape of digital communication and content creation. While these advancements herald a new era of efficiency and creativity, enabling applications ranging from sophisticated writing aids to advanced conversational agents, they simultaneously introduce significant challenges and ethical concerns. In fact, the proliferation of AI-generated texts has raised alarm over issues like the dissemination of misinformation, the facilitation of academic fraud, and the potential erosion of trust in digital content. This underscores the urgent requirement for robust solutions to identify AI-generated content, safeguarding the integrity of information while embracing the benefits of AI advancements.

In this paper, we aim to develop a reliable detection system by participating in the SemEval Task 8 on Machine-Generated Text Detection. This task is notable for its complexity, as it involves Multi-generator, Multidomain, and Multilingual text, making it a highly challenging endeavor. Furthermore, the evaluation is conducted on unseen domains and languages, establishing it as a robust benchmark for evaluating AI text detectors. This requires the model to effectively generalize across different domains and languages. We focus our efforts on the binary detection, which aims to determine whether a text has been generated by an AI or not. To tackle this challenge, we propose a syntactically motivated architecture. Our approach is primarily inspired by the realization that texts generated by AI and humans are semantically similar, as they are derived from comparable topical distributions. Hence, we argue that the distinction between them lies in their syntax and writing style.

Typically, transformer-based text classification relies on information from the last layer for classification. However, our model takes a different approach by dynamically aggregating information from all layers of the transformer (a.k. *multi-layer fusion* Shi et al., 2022). This method is intentionally designed to harness the diverse linguistic information present at various levels of the transformer, as noted in previous studies (Peters et al., 2018; Jawahar et al., 2019; Tenney et al., 2019). These studies reveal the uneven distribution of linguistic features across the transformer's architecture, with syntactic details predominantly in the initial layers and complex semantic information in the deeper layers. By utilizing insights from all layers, our model aims to capture the entire range of linguistic cues, enhancing its capability to accurately differentiate between human and AI-generated content. Additionally, our model moves beyond the standard practice of using just the [CLS] token for classification in BERT-based classifiers. It applies sequence labeling to classify each token in the text as either Human or AI. We believe that

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1166–1171

this approach enables the capture of more complex phrasal structures, which helps in more effectively distinguishing the style and syntax of a text.

Our proposed model obtains competitive performance on the test leaderboard of the shared task subtask A, securing the 2nd best position on binary multilingual detection, using a much smaller model than other approaches often using finetuned LLMs.

2 Related Works

Since the introduction of large-scale pre-trained models like GPT-3, capable of generating high-quality text, the detection of machine-generated text has attracted considerable interest. The most common and straightforward strategy for addressing this task involves training models on a labeled dataset comprising both human and AI-generated text. This approach is utilized by well-known models such as the OpenAI ai text detector and commercial models such as GPTZero (Tian and Cui, 2023). While these models achieve strong in-domain results, they often require labeled datasets from a wide range of sources and domains to achieve generalization. An alternative approach involves zero-shot detectors, which do not necessitate any model training. For example, DNA-GPT (Yang et al., 2024) assesses N-Gram divergence between the continuation distribution of re-prompted text and the original text for making predictions, while Detect-GPT (Mitchell et al., 2023) employs a curvature-based criterion to determine if a passage is generated by a specific LLM.

3 Preliminary study

In this section, we detail a preliminary study that provided essential insights, guiding us towards our final model design.

Motivation Our aim was to assess the efficacy of semantic embeddings, particularly sentence-BERT, in differentiating between machine-generated and human-authored texts. Figure 1 illustrates the embeddings of texts from both humans and various language models, visualized using sentence-BERT embeddings (Reimers and Gurevych, 2019) and UMAP for dimensionality reduction (McInnes et al., 1802).

Analysis The visualization in Figure 1 reveals that texts generated by humans and various language models occupy similar positions in the latent semantic space, with data points from different sources blending together, lacking distinct separation. Given the limited utility of semantic features for discriminating human and ai-generated text, we argue that the key to distinguishing between these texts may lie at the syntactic level.

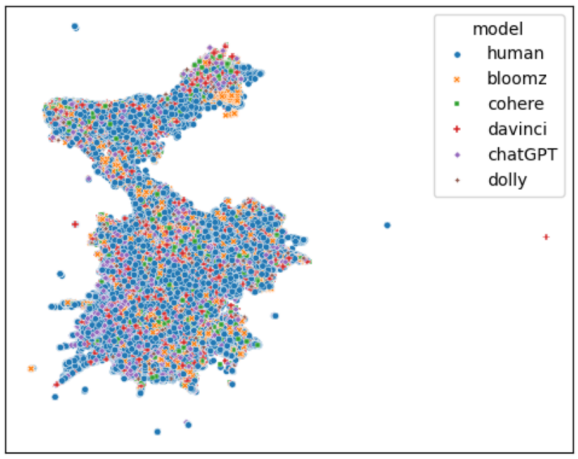


Figure 1: Visualization of UMAP-projected Sentence-BERT embedding of documents generated by human and different large language models

Model Motivated by our analysis, we propose a text classification model that take into account syntactic information. More specifically, our approach introduces two main innovations: 1) the integration of information from all layers of the transformer for classification, referred to as *layer fusion* (Shi et al., 2022). This method leverages the rich linguistic information embedded across the transformer's layers to compute classification scores (Peters et al., 2018; Tenney et al., 2019; Jawahar et al., 2019). 2) The usage of sequence labeling for text classification, which could enhance the model's ability to capture complex phrasal structures, potentially improving its ability to differentiate texts based on style and syntax.

4 Architecture

In this section, we provide a detailed overview of our proposed model's architecture illustrated in Figure 2.

4.1 Representation

Given a text input $X = \{x_1, \ldots, x_N\}$, the model first computes embeddings for each word using a multi-layer pre-trained transformer encoder such as BERT:

$$H = \texttt{transformer}(X) \in \mathbb{R}^{N \times L \times D} \quad (1)$$

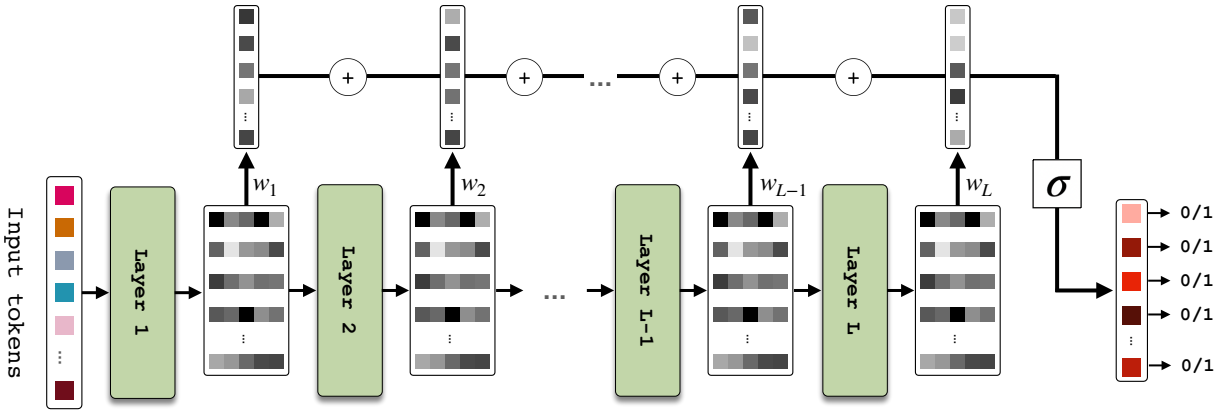


Figure 2: Architecture of Our Proposed Model. A pre-trained transformer receives a sequence of tokens as input and generates token embeddings at each layer. Token scores are computed for each layer, and the final score for each token is derived from the sum of scores across all layers. The probability of each token being AI-generated is determined by applying a sigmoid activation function to its score.

Here, N is the number of words in the input, L is the number of transformer layers, and D is the model dimension.

4.2 Scoring

The model then computes a score for each word, integrating information across all transformer layers, similar to the proposed *multi-layer fusion* by Shi et al. (2022). The score s_i for a word at position i is computed as follows:

$$s_i = \sum_{l=1}^{L} \boldsymbol{w}_l^\top \boldsymbol{h}_i^l \in \mathbb{R} \tag{2}$$

In this equation, $\boldsymbol{h}_i^l \in \mathbb{R}^D$ represents the embedding of the i-th word at the l-th layer. $\boldsymbol{w}_l \in \mathbb{R}^D$ is a learned weight vector specific to layer l. This scoring mechanism allows the model to weigh the contributions of different layers differently for each token, potentially emphasizing certain linguistic features over others.

4.3 Classification

For the classification, we employ a sequence labeling approach, where each word is classified based on its computed score. For this, a sigmoid function is applied to convert the token scores into probabilities, and a threshold is used to make a binary decision:

$$y_i = \begin{cases} 1 & \text{if } \sigma(s_i) > 0.5, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This step results in a binary classification for each word, indicating its belonging to the positive class. Finally, the classification of the entire sentence is determined by averaging these binary decisions:

$$y = \frac{1}{N} \sum_{i=1}^{N} y_i \tag{4}$$

This average represents the probability of the sentence belonging to the positive class, synthesizing the word-level classifications into an overall sentence-level prediction. Finally, given an input X, we consider it as being machine-generated if its computed probability is superior to 0.5, i.e $y > 0.5$.

4.4 Training

To train our model, we focus on maximizing the likelihood of the correct label for each token by minimizing the binary cross-entropy loss at the token level. The binary cross-entropy loss for an input text of length N can be formulated as follows:

$$\mathcal{L} = -\sum_{i=1}^{N} \left(y^* \log(p_i) + (1 - y^*) \log(1 - p_i)\right) \tag{5}$$

Here, y^* represents the true label of the input (1 for human-generated and 0 for AI-generated), p_i denotes the predicted probability of the i-th token being human-generated (computed by applying the sigmoid function to the score s_i).

5 Experimental setup

5.1 Data

For our experiments, we used the dataset provided at SemEval-2024 Task 8, more details can be found in the task description (Wang et al., 2024a). It is

based on the benchmark M4 dataset (Wang et al., 2024b), which is a large-scale multi-generator, Multi-domain, and Multi-lingual corpus containing human-written and machine-generated texts. The machine-generated texts were produced by prompting several LLMs, including ChatGPT, textdavinci-003, Cohere, Dolly-v2 and BLOOMz from different sources such as Wikipedia, WikiHow, Reddit, arXiv, PeerRead for English, Baike and Web question answering for Chinese, news for Urdu, RuATD for Bulgarian and news for Indonesian.

5.2 Hyperparameters

In our experiments, we utilized the `xlm-roberta-large` model as the backbone for our architecture. The model was trained with a batch size of 12 across a maximum of 2 epochs, as we found that training further harms the validation results. More specifically, we observed that while training longer always improves in-domain performance measured on a held-out subset of the training set, it harms performance on out-of-domain validation (Kumar et al., 2022). We hypothesize this is due to overfitting on in-domain data, making long training harms the generalization of the model. Due to this, we evaluated our model on the out-of-domain validation set every 500 gradient steps and kept the best-performing model for testing. We employed different learning rates for the backbone (pre-trained transformer model) parameters and the added projection parameters: the learning rate for the backbone was set to `1e-5`, and the learning rate for the projection weights (randomly initialized) was set higher at `3e-4`. This distinction allows for delicate fine-tuning of the pre-trained model (to not distort the pre-trained representation too much), while more aggressively updating the newly introduced parameters to adapt to the task-specific features. During training, we use a maximum sequence length of 128 subwords to allow faster training, but we compute test set prediction using the maximum size of 512 tokens. The experiments were conducted with a runtime limit of 2 hours and 30 minutes for each experiment, utilizing an NVIDIA V100 GPU.

5.3 Other approaches

In this section, we provide an overview of the methodologies adopted by participants based on their description[1] in the shared task (Wang et al., 2024a). The baseline approach involved fine-tuning an XLM-Roberta-base model specifically for this task. The team *USTC-BUPT* presented the top-performing system, where English texts were processed using the Llama-2-70b model to generate average embeddings. These embeddings were then classified using a two-stage CNN. For texts in languages other than English, they treated classification as a next-token prediction task utilizing the mT5 model. Another notable participant, the *KInIT* team, employed an ensemble strategy that combined fine-tuned large language models (LLMs), including Mistart and Falcon, with zero-shot statistical methods to improve performance. Lastly, the *L3i++* team opted to fine-tune a LLaMA-2–7b model for the task. In comparison to the top-performing participants, only our approach uses small-scale transformer models.

[1]Note that we do not have access to entire articles.

Rank	Team	Accuracy (%)
1	USTC-BUPT	95.9
2	FI Group (*Ours*)	95.8
3	KInIT	95.0
4	priyansk	93.8
5	L3i++	92.9
–	*Baseline*	80.9

Table 1: Test leaderboard results.

6 Results

6.1 Test leaderboard results

Table 1 shows the top 5 scores from the leaderboard obtained using the test dataset, which includes domains and languages never seen during training.

Our team achieved the second-highest score, with an accuracy of 95.8%, narrowly trailing the top system by only 0.1%. There were 69 participants in the multilingual track in subtask A, out of a total of 159 participants across all SemEval-2024 Task 8.

6.2 Ablation study

In this section, we conduct an ablation study to examine the impact of various components of our model, including the backbone, layer fusion, and sequence labeling. The outcomes of this analysis are reported in Table 2.

Results Regarding the backbone, our findings indicate that `XLM-R-large` achieves better per-

Model	Accuracy (%)	F1 (%)
Ours (`XLM-R-base`)	87.3	87.1
Ours (`XLM-R-large`)	87.6	87.5
- w/o sequence labeling	81.2	81.1
- w/o layer fusion	78.1	77.4
Baseline (`XLM-R-base`)	75.0	–

Table 2: Ablation performance on the validation set. We perform ablation of our proposed model to see the influence of sequence labeling and layer fusion.

formance than `XLM-R-base`, suggesting that our method scales effectively. Moreover, our analysis reveals that both sequence labeling and layer fusion significantly contribute to the model's performance. Specifically, omitting sequence labeling—which involves aggregating the scores of the CLS token across layers—results in a 6-point decrease in accuracy. Similarly, excluding layer fusion leads to a more pronounced decline, with over a 10-point drop in F1 score and a 9-point decrease in accuracy. These findings underscore the critical roles that token-level prediction and layer fusion play in enhancing the overall effectiveness of our model.

6.3 Learned Weight Analysis

Motivation Figure 3 visualizes the norm of the learned weight vector for each layer of our model, denoted as w_l in equation 2. We hypothesize that the magnitude of these projection weights reflects the significance of each layer in contributing to the final prediction, with higher weights suggesting a more substantial influence on the token scores.

Analysis The Figure 3 indicates that layer 0, the embedding layer, has the lowest norm value. Given that this layer does not incorporate contextual information, its minimal contribution suggests that mere word appearance is insufficient for determining whether a text is produced by a human or an AI, aligning with the findings of Gallé et al. (2021). Interestingly, layer 24, which is the final layer, also shows a relatively low norm value. This observation resonates with analyses indicating that the last layer tends to be rich in semantic content yet sparse in syntactic details. We believe this explains the lower norm value for the last layer, as semantic aspects alone are inadequate for distinguishing between human and AI writing. Conversely, the highest norm values are predominantly found in layers 3 to 6 and 20 to 22, suggesting these layers

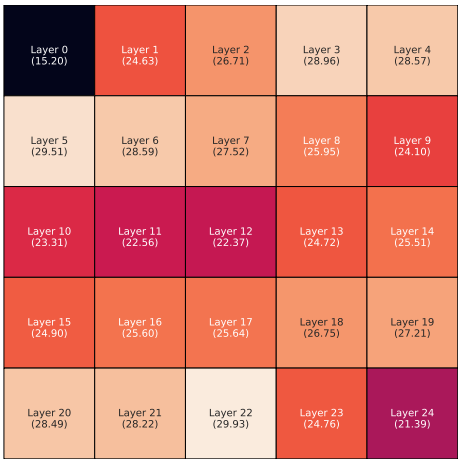


Figure 3: Norm (L1) of the weight vector w_l for each layer in our model.

play a pivotal role in the model's decision-making process.

7 Conclusion

In this paper, we presented our system submitted to SemEval-2024 Task 8 for detecting human-written and machine-generated text, achieving 2nd place for subtask A on multilingual texts. Our system relies on a hierarchical fusion strategy that adaptively fuses representations from transformer's layers, with a focus on syntactic rather than semantic information. By leveraging syntactic features, particularly through sequence labeling, we captured more phrasal structures of text, thereby enhancing our ability to distinguish text styles and syntax. Our system achieved robust performance across diverse unseen domains and languages, demonstrating its adaptability and generalization capability, notably considering that we used a smaller model compared to other proposed systems often reliant on fine-tuned LLMs.

References

Matthias Gallé, Jos Rozen, Germán Kruszewski, and Hady Elsahar. 2021. Unsupervised and distributional detection of machine-generated text. ArXiv: 2111.02878 [cs.CL].

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. Fine-

Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *International Conference on Learning Representations*.

Leland McInnes, John Healy, and James Melville. 1802. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv 2018. *arXiv preprint arXiv:1802.03426*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting Contextual Word Embeddings: Architecture and Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-Networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing*. Association for Computational Linguistics.

Han Shi, JIAHUI GAO, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen M. S. Lee, and James Kwok. 2022. Revisiting over-smoothing in BERT from the perspective of graph. In *International conference on learning representations*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Edward Tian and Alexander Cui. 2023. GPTZero: Towards detection of AI-generated text using zero-shot and supervised methods.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th international workshop on semantic evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text. In *The Twelfth International Conference on Learning Representations*.

Team Innovative at SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection

Surbhi Sharma
Purdue University
surbhisharma9099@gmail.com

Irfan Mansuri
SWE Qualcomm
iffyaiyan@gmail.com

Abstract

With the widespread adoption of large language models (LLMs), such as ChatGPT and GPT-4, in various domains, concerns regarding their potential misuse, including spreading misinformation and disrupting education, have escalated. The need to discern between human-generated and machine-generated text has become increasingly crucial. This paper addresses the challenge of automatic text classification with a focus on distinguishing between human-written and machine-generated text. Leveraging the robust capabilities of the RoBERTa model, we propose an approach for text classification, termed as RoBERTa hybrid, which involves fine-tuning the pre-trained Roberta model coupled with additional dense layers and softmax activation for authorship attribution. In this paper, we present an approach that leverages Fabien et al. (2020) Stylometric features, hybrid features, and the output probabilities of a fine-tuned RoBERTa model. Our method achieves a test accuracy of 73% and a validation accuracy of 89%, demonstrating promising advancements in the field of machine-generated text detection. These results mark significant progress in the domain of machine-generated text detection, as evidenced by our 74th position on the leaderboard for Subtask-A of SemEval-2024 Task 8.

1 Introduction

SemEval-2024 Task 8 Wang et al. (2024) is centered on the detection of machine-generated text across multiple generators, domains, and languages. This detection is crucial for mitigating the risks associated with the potential misuse of large language models (LLMs), which have advanced capabilities in generating multilingual human-like texts. In this task, the goal is to differentiate between machine-generated and human-authored texts, addressing concerns regarding the authenticity and trustworthiness of textual content in various contexts and languages.

The rapid advancement of deep learning technologies has ushered in a new era where the boundaries between human-generated and machine-generated artifacts are increasingly blurred. This evolution is epitomized by the emergence of Deepfakes, which convincingly mimic genuine human actions, and the widespread adoption of Natural Language Generation (NLG) systems, particularly those leveraging neural language models. These developments have led to the creation of neural texts that bear striking resemblances to human-authored content, posing significant challenges in distinguishing between the two.

Traditionally, Authorship Attribution Uchendu et al. (2020) within the realm of Natural Language Processing (NLP) focused on accurately attributing text to its true human author. However, with the advent of Neural Language Generation (NLG) techniques Uchendu et al. (2023) capable of producing human-quality open-ended texts, the attribution landscape has expanded to encompass authorship by humans, machines, or a combination thereof. As the quality of machine-generated texts continues to improve, the lines between human and machine-generated text become increasingly indistinct, exacerbating the challenge of differentiation.

Moreover, the potential for misuse of these technologies, including the generation of misinformation, fake reviews, and political propaganda at scale Uchendu et al. (2023), underscores the critical need for effective methods to discern neural texts from human-authored content—a problem known as Neural Text Detection (NTD) Uchendu et al. (2023), which is a sub-problem of the broader authorship attribution domain.

In this paper, we present an approach named RoBERTa hybrid, tailored to address the challenge of distinguishing between human and machine-generated texts. This method utilizes the fine-tuning of a pre-trained RoBERTa language model, enhanced with additional layers for text classifica-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1172–1176

tion, in evaluating the performance of pre-trained language models for text differentiation tasks. We specifically target the task of automated detection of machine-generated text, recognizing the growing importance of discerning between human-authored and artificially generated content in today's digital landscape.

In this paper, we present an approach named RoBERTa hybrid, tailored to address the challenge of distinguishing between human and machine-generated texts. This method utilizes the fine-tuning of a pre-trained RoBERTa language model, enhanced with additional layers for text classification, in evaluating the performance of pre-trained language models for text differentiation tasks. We specifically target the task of automated detection of machine-generated text, recognizing the growing importance of discerning between human-authored and artificially generated content in today's digital landscape. We secured the rank 74 on the leaderboard for SubTasK-A

2 Related Work And Background

Subtask A of Task-8 in the SemEval challenge Wang et al. (2024) utilized a monolingual dataset, focusing on distinguishing between human-written and machine-generated text. The dataset primarily employed English as its medium. Each instance in the dataset is labeled as either 1 (indicating machine-generated text, specifically generated by Chat-GPT) or 0 (indicating human-written text).

The Subtask A dataset consists of three subsets: training, development, and testing datasets. The training dataset contains 119,757 samples, while the development dataset comprises 5,000 samples. The testing dataset includes 34,272 samples. In the training dataset, there are 63,351 samples labeled as class 0 and 56,406 samples labeled as class 1. Conversely, the development dataset consists of 2,500 samples, all of which are labeled.

Within the dataset, there are columns denoted as "model" and "source." The "model" column specifies which model generated a particular text, while the "source" column indicates the origin of the text.

Previous research has explored various approaches to authorship attribution (AA), aiming to accurately identify the authors of texts, which is crucial in fields such as forensic linguistics, plagiarism detection, and content analysis. Traditional methods have relied on Stylometric features, which capture the distinctive writing style of individual authors based on linguistic patterns and characteristics. However, recent advancements in natural language processing (NLP) have introduced transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) that have demonstrated state-of-the-art performance across a range of NLP tasks. Fabien et al. (2020) showcased the effectiveness of BERT for text classification tasks, highlighting its ability to extract semantic and syntactic information from text. However, there has been a lack of systematic exploration into the performance of fine-tuned pre-trained language models, specifically for authorship attribution. The introduction of BertAA addresses this gap by fine-tuning BERT with a dense layer and softmax activation specifically for authorship attribution. The incorporation of BERT allows for the utilization of semantic and syntactic information encoded in text representations, potentially improving the accuracy of authorship attribution systems. Furthermore, BertAA integrates Stylometric features, which capture lexical and structural characteristics of text, and hybrid features, which combine character-level n-grams, enhancing the model's ability to capture both content-related and stylistic aspects of authorship.

However, Stylometric classifiers encounter challenges when tasked with accurately determining the authorship of human versus neural texts. Uchendu et al. (2023) noted that certain Stylometric classifiers were surpassed in performance by deep learning-based models. Furthermore, research findings, as cited in Schuster et al. (2020), revealing Stylometry's inability to identify neural misinformation underscore the necessity for alternative methodologies to address the Authorship Attribution (AA) task within the context of Neural Text Generation (NTD). Consequently, researchers have increasingly embraced and refined deep learning-based approaches for distinguishing between neural and human-generated text. These approaches can be further classified into three main categories: Glove-based, Energy-based, and Transformer-based Attribution models.

Language models often exhibit a lack of syntactic and lexical diversity, characterized by the repetition of the same expressions and a limited use of synonyms and references. This behavior Fröhling and Zubiaga (2021) can be approximated using named entities (NE) and properties of coreference chains, along with shifts in part-

of-speech (POS) distributions between human and machine-generated text. Features based on NE-tags, coreference chains, and POS distributions Fröhling and Zubiaga (2021) can effectively capture the differences in syntactic and lexical diversity Gehrmann et al. (2019) between human and machine-generated text.

Repetitiveness: Machine-generated text is prone to repetitiveness Holtzman et al. (2019), often overusing frequent words and exhibiting highly parallel sentence structures. Features such as the share of stop-words, unique words, and words from "top-lists" can highlight the lack of diversity in machine-generated text. Additionally, measures of n-gram overlap in consecutive sentences can reveal patterns of lexical and syntactic repetition, further distinguishing between human and machine-generated text.

Lack of Coherence: A significant challenge in machine-generated text is the lack of coherence Holtzman et al. (2019), particularly over longer sentences and paragraphs. Coherence can be assessed through the development of entities and the tracking of their appearance and grammatical roles across the text. Features based on entity grids and transition frequencies Badaskar et al. (2008) between consecutive sentences can capture the coherence or lack thereof in machine-generated text.

By incorporating these features into automated detection methods, researchers aim to develop robust and accessible tools for distinguishing between human and machine-generated text, thereby mitigating the risks associated with language model abuse.

3 System Overview

The experiments conducted encompassed Subtask A within the monolingual track. Subtask A posed a binary classification challenge, aimed at discriminating between human-generated text and text produced by the Machine (ChatGPT).

In addressing Subtask A, a Stylometric classifier was developed to exploit diverse stylistic attributes, encompassing text length, word count, average word length, count of short words, proportion of digits and capital letters, frequencies of individual characters and digits, hapax-legomena (a measure of text richness), and the frequency of 12 punctuation marks. These Stylometric features were employed in training a Logistic Regression model.

Furthermore, hybrid features, incorporating the 100 most frequent character-level bi-grams and tri-grams, were integrated. Logistic Regression was applied for classification using these hybrid features as well.

The ultimate model adopted a hybrid strategy, whereby output probabilities from the RoBERTa classifier, the Stylometric classifier, and the hybrid features classifier were concatenated. This concatenated output underwent classification using an additional Logistic Regression model. We chose RoBERTa due to its robust performance in natural language understanding tasks, its ability to handle a wide range of text data, and its pre-training on large-scale corpora, which helps capture nuanced linguistic patterns.

To refine the RoBERTa hybrid model, class probabilities derived from the Stylometric features and those obtained from fine-tuning the RoBERTa model were concatenated separately for both the training and test datasets. Additionally, the probabilities derived from training a Logistic Regression model on hybrid features were integrated into the hybrid model.

3.1 Stylometric Features Extraction

- Length of text: Count the number of characters or tokens.
- Number of words: Total word count.
- Average word length: Average length of words.
- Number of short words: Count of words below a certain threshold.
- Proportion of digits and capital letters: Ratio of digits and capitals to total characters.
- Individual letter and digit frequencies: Count of each letter and digit.
- Hapax-legomena: Words occurring only once.
- Frequency of punctuation marks: Count of specific punctuation marks.

3.2 Hybrid Features Extraction

- Character-level n-grams: Extract the 100 top frequent character-level bi-grams and tri-grams in the text.

3.3 Logistic Regression Model

- In logistic regression, the input features are linearly combined, and the result is passed through the logistic function (also known as

the sigmoid function) to obtain the probability of the positive class.

- Mathematically, the logistic regression model can be represented as:

$$P(y = 1|x) = sigmoid(w^T \cdot x + b)$$

where $P(y = 1|x)$ is the probability of the positive class given the input features x, w represents the weight vector, b is the bias term, and $sigmoid$ is the logistic function.

Hyperparameters:

- **Penalty**: This hyperparameter controls the regularization strength, with options typically including L1 (Lasso) or L2 (Ridge) regularization.
- **Tolerance**: It determines the stopping criteria for the optimization algorithm, specifying the tolerance for the change in the loss function between iterations.
- **Maximum Iterations**: This sets the maximum number of iterations allowed for the optimization algorithm to converge.
- **Intercept**: A boolean parameter indicating whether to include an intercept term in the model.

These hyperparameters are crucial for controlling the model's complexity, preventing overfitting, and optimizing performance. They are typically tuned using techniques like grid search or cross-validation to find the best combination for the given dataset and task.

Model	Parameter	Value
Hybrid feat.	Char. N-grams	(2,3)
LR	Penalty	l2
	Tolerance	0.0001
	Cost	1.0
	Max Iterations	100
	Intercept	True
RoBERTa	Max Iterations	100
	Intercept	True
	Config Epochs	1 to 5
	Input token length	512

Table 1: Parameters of the experiments.

4 Experimental Setup and Evaluation Results

Experimental Design: In our experimental setup, we fine-tuned the RoBERTa model over 5 epochs using the training dataset. To ensure model robustness and prevent overfitting, we utilized a validation dataset consisting of 80% of the training data and 20% of the testing data.

Our approach involved creating a hybrid model, which integrated class probabilities from three classifiers: Stylistic classifier, hybrid classifier, and RoBERTa classifier. These probabilities were concatenated and passed through a Logistic Regression layer for training.

To assess the efficacy of our model, we evaluated its performance using the accuracy metric.

Classifier	Accuracy (%)
Stylometric classifier	49
Hybrid Features Classifier	58
RoBERTa + Style Classifier	73
Hybrid Classifier (RoBERTa + Style + Hybrid)	73

Table 2: Accuracy Results on the Test Dataset

Evaluation of Results: The accuracy results on the test dataset are summarized in Table 2. We observe varying performance among different classifiers. The Hybrid Classifier (RoBERTa + Style + Hybrid) achieved the highest accuracy of 73%, outperforming both the Stylometriv Classifier (49%) and the Hybrid Features Classifier 58%. However, (RoBERTa + Style) based Classifier too resulted in 73% This indicates that incorporating RoBERTa-based representations along with Stylometric and hybrid features significantly improved the model's ability to classify text accurately.

The superior performance of the Hybrid Classifier can be attributed to its utilization of RoBERTa, a transformer-based model known for its ability to capture rich contextual information from text. By leveraging RoBERTa's representations along with Stylometric and hybrid features, the Hybrid Classifier achieved a more comprehensive understanding of the input text, leading to better classification accuracy.

On the other hand, the Stylometric Classifier and the Hybrid Features Classifier exhibited lower

accuracies compared to the Hybrid Classifier. This could be due to their reliance on a narrower set of features for classification, which may not capture the full complexity of the input text.

5 Conclusion

In this paper, we have presented an innovative approach, termed the RoBERTa hybrid model, for the task of detecting machine-generated text. Leveraging the robust capabilities of the Roberta model, we fine-tuned it coupled with additional dense layers and softmax activation for authorship attribution. Our method incorporates a hybrid of Stylometric features, character-level n-grams, and the output probabilities of a fine-tuned Roberta model, achieving significant advancements in machine-generated text detection.

Experimental results demonstrate the effectiveness of our approach, with a validation accuracy of 89% and a test accuracy of 73%. Although these results do not surpass the baseline methods, they highlight the potential of our approach in addressing the challenges posed by machine-generated text across diverse domains.

Moving forward, our work opens avenues for further research in enhancing the accuracy and robustness of machine-generated text detection systems. Future efforts may focus on exploring additional feature representations, optimizing model architectures, and addressing the challenges posed by monolingual machine-generated text. Our efforts in enhancing machine-generated text detection we have tried to contribute to the broader objective of safeguarding the integrity and credibility of online content.

Acknowledgements

We extend our heartfelt appreciation to all contributors to this research. Our Supervisor has provided invaluable insights and feedback, enriching the quality of our work. We extend special gratitude to the reviewers for their constructive comments and suggestions, which greatly enhanced the paper. The first author led the experimental design and paper writing process, while the second author with my guidance has contributed exclusively to proofreading and paper writing, not involved in the experimental design.

References

Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Maël Fabien, Esaú Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. Bertaa: Bert fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7:e443.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Tal Schuster, Roei Schuster, Darsh J Shah, and Regina Barzilay. 2020. The limitations of stylometry for detecting machine-generated fake news. *Computational Linguistics*, 46(2):499–510.

Adaku Uchendu, Thai Le, and Dongwon Lee. 2023. Attribution and obfuscation of neural text authorship: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 25(1):1–18.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

EURECOM at SemEval-2024 Task 4: Hierarchical Loss and Model Ensembling in Detecting Persuasion Techniques

Youri Peskine and **Raphael Troncy** and **Paolo Papotti**
EURECOM, France
firstname.lastname@eurecom.fr

Abstract

This paper describes the submission of team EURECOM at SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. We only tackled the first sub-task, consisting of detecting 20 named persuasion techniques in the textual content of memes. We trained multiple BERT-based models (BERT, RoBERTa, BERT pre-trained on harmful detection) using different losses (Cross Entropy, Binary Cross Entropy, Focal Loss and a custom-made hierarchical loss). The best results were obtained by leveraging the hierarchical nature of the data, by outputting ancestor classes and with a hierarchical loss. Our final submission consist of an ensembling of our top-3 best models for each persuasion techniques. We obtain hierarchical F1 scores of 0.655 (English), 0.345 (Bulgarian), 0.442 (North Macedonian) and 0.177 (Arabic) on the test set.

1 Introduction

Online misinformation is a complex research topic. It can appear in many shape and forms (text, images, videos, etc.), within different contexts (political debates, news articles, social media posts, etc.). As memes generate high engagement on social media, they are also used for disinformation campaign by exploiting rhetoric persuasion. This year's "SemEval-Task4: Multilingual Detection of Persuasion Techniques in Memes" task aims at detecting named persuasion techniques in memes. The full overview of the task and its sub-tasks are detailed in (Dimitrov et al., 2024).

As a brief summary, SemEval-2024 Task 4 consist of detecting persuasive techniques in Memes. The task breaks down into 3 sub-tasks; sub-task 1 use the textual content of the meme to detect the persuasion techniques, sub-task 2a use the whole image and text to detect the persuasive techniques, while sub-task 2b only consist of binary detection. In sub-task 1, a total of 20 persuasion techniques are used. In this work, we only describe our solution to tackle this sub-task 1.

Our approach consists of an ensembling model of our top-3 best models for each persuasion techniques. In our experiments, we reached the best results leveraging the hierarchical nature of the data, with hierarchical loss, and outputting ancestor classes. Our method can be reproduced using the code at https://github.com/D2KLab/semeval-2024-task-4.

2 System Description

In this section, we describe the system used in our submission. We also present approaches that were considered but not kept in our final submission.

2.1 Models

We experimented with multiple transformer-based models to tackle persuasion detection in the textual content of the memes.

- **BERT** (Devlin et al., 2019): First introduced in 2018, this model is based on the bidirectional transformer encoder architecture (Vaswani et al., 2023) trained with masked language model and next sentence prediction tasks.
- **BERT-HarMe**[1]: This model is a fine-tuned version of BERT on multiple datasets[2] (Kiela et al., 2021; Suryawanshi et al., 2020) about harmful/hateful speech in memes.
- **RoBERTa** (Liu et al., 2019): This model changes the BERT pre-training approach, making it more robust.
- **AlBERT** (Lan et al., 2020): AlBERT focuses on reducing the number of parameters

[1] https://huggingface.co/limjiayi/bert-hateful-memes-expanded
[2] https://github.com/di-dimitrov/harmeme

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1177–1182

Dataset	Size
SemEval-2024 Train	7000
SemEval-2021 Train+Validation+Dev	951
PTC (sampled)	427

Table 1: Datasets considered for training our models.

of BERT to increase the training speed and lower memory requirements.

- **DistilBERT** (Sanh et al., 2020): This model uses knowledge distillation during pre-training to reduce the size of BERT.
- **DeBERTa** (He et al., 2021): DeBERTa improves on BERT and RoBERTa by introducing a disentangled attention mechanism and an enhanced mask decoder.

2.2 Datasets

In this task, we use multiple training datasets. We experimented adding the train, validation and dev sets from SemEval-2021 Task 6 (Dimitrov et al., 2021) and the PTC corpus (Da San Martino et al., 2020) to the training data. Table 1 shows the datasets and their respective sizes.

- **SemEval-2021 Task 6**: This dataset also annotates memes with regards to the same 20 persuasion techniques. The train, validation and dev sets are appended to the training set of this task without any modification.
- **PTC Corpus**: This dataset contains news articles annotated at the span level with regards to 18 propaganda techniques. We first split the articles into sentences and transfer the span-level label to sentence-level. In this dataset, some labels are the same as this year's task, and can be aligned in a straightforward manner. However, when propaganda labels are different, they often correspond to multiple persuasion techniques. To align these labels, we add all the corresponding persuasion techniques valid for the propaganda. We only appended sentences that contain a propaganda technique to the training set of this task (around 5% of the total number of sentences).

2.3 Outputting ancestor classes

In this task, the goal is to detect the 20 persuasion techniques, but they appear in a hierarchical framework. The official metrics of the challenge are hierarchical F1 (**F1H**), hierarchical precision (**PreH**) and hierarchical recall (**RecH**), which all take into consideration the hierarchical nature of the data. Since ancestor nodes are inherently outputted when detecting child nodes, we also tried to directly detect the ancestor classes. This raises the number of classes to 28 (instead of 20). Thus, the ancestor node can still be outputted even if it's child node has not been detected, resulting in better performing models.

2.4 Losses

We also experimented with different training losses, which address multiple aspects of the data. For example, balancing the classes misrepresentation in the data with class weights, or using hierarchical loss to reflect the hierarchical nature of the data.

- **Binary Cross Entropy (BCE) Loss**: This loss computes BCE losses for each class, weighted with the inverse frequency of its label, and sum them. This loss requires the output layer to have the size of number of classes.
- **Cross Entropy (CE) Loss**: We used 20 different CE losses for each class, weighted according to the inverse frequency of each label. Each loss computes the performance of the model at detecting a specific class. The final loss is the sum of the 20 losses. This loss requires the output layer to have twice the size of number of classes.
- **Focal Loss (FL)** (Lin et al., 2020): This loss addresses class imbalance by down-weighting the loss assigned to well-classified examples. We used the implementation proposed by (Edgar et al., 2020). This loss requires the output layer to have the size of number of classes.
- **Custom Hierarchical Loss (HL)**: In order to reflect the hierarchical nature of the data, we implemented a custom hierarchical loss function. This function uses max pooling on logits x^c from children classes of the same ancestor a (e.g. Name Calling, Doubt, Smears, Reductio ad Hitlerum and Whataboutism are all children of the Ad hominem ancestor). The newly created logit correspond to the output of the model on the corresponding ancestor. Thus, we can compute the BCE Loss between

this output and the true label y^a of the ancestor. We can iterate by max-pooling all the logits in the next ancestor. Note that logits can correspond to children or ancestor classes (e.g. the Logos ancestor pools the logits of Justification, Repetition, Intentional Vagueness, and Reasoning, even though the logits of Justification and Reasoning are also pooled from other child classes). We then sum all these BCE losses together, which measure how well the model performs to detect the ancestor rather than each persuasion techniques. Before summing this loss to the original classification loss of the techniques (CE, BCE or FL), we apply a normalization factor α. In practice, we found best results when α is equal to 0.5. Equations 1 and 2 describe the computation of this loss. $\mathcal{A}$ describes the ensemble of all ancestor techniques.

$$\mathcal{L}_{HL} = \mathcal{L}_{CE,BCE} + \alpha \cdot \sum_{a \in \mathcal{A}} \mathcal{L}^a_{BCE} \quad (1)$$

$$\mathcal{L}^a_{BCE} = y^a \cdot log\sigma(\max(\{x^c\}_{c \in child(a)})) + (1 - y^a) \cdot log(1 - \sigma(\max(\{x^c\}_{c \in child(a)}))) \quad (2)$$

2.5 Data augmentation

Some persuasion techniques have very little training data available in the datasets. We tried generating new samples for the bottom 5 classes with different methods.

- **Round Translation**: We translated every sample in French and translated them back to English. This can generate new sentences similar to the original ones. However, this new data is very limited and will not be varied.
- **GPT-4-Turbo Generation** (et al., 2023): We used GPT-4-Turbo to generate completely new sentences corresponding to a persuasive technique. As showed in (Peskine et al., 2023), definitions of the class label have a significant impact in the performance of GPT models. We provided the definition of the persuasive technique provided by the organizers[3] in the system prompt, along with 5 randomly selected samples. We then used few-shot prompt technique with 5 more randomly selected samples, and finally asked the model to generate a new sentence. We generated two sets of 30 and 50 examples for five classes. For reproducibility measures, the full prompt is available in Appendix A.

2.6 Training process

For training our models, we use the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 2e-5, and a weight decay of 0.01. We also use a ReduceLROnPlateau Learning rate scheduler, reducing the earning rate by a factor of 0.7 if results have not improved in 4 epochs. Most experiments are done on 10 epochs, saving the best model (according to F1H) on the validation set. We also experimented with freezing the first few layers of the pre-trained BERT-based model to keep its acquired knowledge when trained on massive amount of data.

2.7 Ensembling

We trained many models according to different combinations of the previous parameters. Our final submission consists of a majority voting among the top-3 models for each persuasion technique evaluated on the dev set and according to the F1-score. These models are not necessarily the best models overall according to hierarchical F1, but demonstrate effectiveness in detecting specific persuasion technique. We also perform majority-voting on ancestor classes with models that output them (Section 2.3).

3 Results

We share our results on the dev set provided by the organisers in Table 2. These results show the performance of some single models as well as the performance of the ensembling used in the final submission. Table 4 shows the performance of each class on the dev set, using the ensembling model for classification. Table 3 shows the results of our final submission on the 4 test languages: English, Bulgarian, North Macedonian and Arabic. We translate non-English languages using `py-googletrans`[4] to English in order to run our models and obtain the predictions. We would like to note that our official submission for the Arabic language was incorrect, due to Arabic-to-English translation errors on our

[3] https://propaganda.math.unipd.it/semeval2024task4/definitions.html

[4] https://github.com/ssut/py-googletrans

Model	Data	Classes	Loss	F1H	PreH	RecH
BERT	2024	20	CE	0.612	0.603	0.621
BERT	2024+2021	20	BCE	0.623	0.561	**0.700**
BERT	2024+2021	28	HL	**0.640**	0.626	0.654
BERT	2024+2021+PTC	28	HL	0.633	**0.647**	0.618
BERT	2024+2021	28	FL	0.629	0.638	0.620
BERT	2024+2021	20	FL	0.611	0.635	0.588
BERT	2024	28	CE	0.629	0.612	0.646
RoBERTa	2024+2021	20	CE	0.619	0.610	0.628
RoBERTa	2024+2021	28	CE	0.631	0.610	0.653
BERT-HarMe	2024+2021	20	CE	0.625	0.599	0.652
BERT-HarMe	2024+2021	28	CE	0.639	0.651	0.627
BERT-HarMe	2024+2021	28	HL	0.634	0.634	0.634
BERT-HarMe	2024+GPT-augmented	28	CE	0.634	0.605	0.666
AlBERT	2024+2021	20	CE	0.604	0.600	0.607
DeBERTa	2024+2021	20	CE	0.617	0.617	0.618
DistilBERT	2024+2021	20	CE	0.602	0.622	0.584
Ensembling	Top-3 best models			**0.675**	**0.650**	**0.702**

Table 2: Results on the dev set of some of the models we tried. Other models with different combination of parameters are used in the ensembling and not showed here due to space, but obtain similar performances.

Language	F1H	PreH	RecH
English	0.655	0.628	0.685
Bulgarian	0.345	0.367	0.325
North Macedonian	0.442	0.520	0.384
Arabic	0.177	0.343	0.119
Arabic (unofficial)	0.439	0.369	0.544

Table 3: Results on the test set with our ensembling model, translating non-English languages to English.

end. We corrected the error and also show the performance of the model, albeit being an unofficial result.

4 Discussion

Model-wise, our best results were obtained using BERT, RoBERTa and BERT-HarMe. We ultimately did not use any of AlBERT, DeBERTa and DistilBERT models in our final submission as those were not in any top-3 best performing models of any persuasion techniques. The BERT-HarMe models were the best-performing on the detection of 'Slogans', 'Appeal to Authority', 'Flag-waving', 'Appeal to fear/prejudice', 'Black-and-white Fallacy/Dictatorship', 'Thought-terminating cliché', 'Presenting Irrelevant Data (Red Herring)', 'Glittering generalities (Virtue)', 'Doubt', 'Logos', 'Justification' and 'Distraction' classes. RoBERTa models were the best-performing for 'Repetition', 'Bandwagon', 'Ethos'.

We also noticed a slight performance increase by adding the 2021 dataset during training, which was not necessarily true when adding the PTC corpus. This is probably due to the fact that the PTC Corpus is about news articles and not memes. Our data-augmentation experiments on round-translation did not improve the results at all, while the GPT-4-Turbo augmentation experiments provided a very slight boost, but not for the augmented classes.

The hierarchical nature of the task and the evaluation metrics were reflected in the results, as most of our best performing models are outputting 28 classes by including the ancestors and/or are trained with Hierarchical Loss (**HL**). However, best models at detecting 'Causal-Oversimplification' are using BCE Loss.

We can see in Table 4 that some persuasive techniques are easier to detect than others. For example, 'Appeal to authority' seems to be the easiest class to detect, and 'Obfuscation, Intentional vagueness, Confusion' the hardest. Training data seems to lightly correlate with performance results, with some strong outliers like 'Smears' underperforming comparing to it's high number of training samples, and 'Bandwagon' over-performing. As for the ancestor classes, the highest-level 'Logos', 'Ethos' and 'Pathos' have the highest performance, while those composed of the hardest per-

Technique	F1H
Repetition	0.516
Obfuscation	0.000
Slogans	0.495
Bandwagon	0.583
Appeal to authority	0.891
Flag-waving	0.623
Appeal to fear/prejudice	0.425
Causal Oversimplification	0.304
Black-and-white Fallacy	0.549
Thought-terminating cliché	0.330
Straw Man	0.286
Red Herring	0.182
Whataboutism	0.442
Glittering generalities (Virtue)	0.562
Doubt	0.437
Name calling/Labeling	0.617
Smears	0.583
Reductio ad hitlerum	0.526
Exaggeration/Minimisation	0.492
Loaded Language	0.682
Logos	0.773
Reasoning	0.552
Justification	0.727
Simplification	0.496
Distraction	0.389
Ethos	0.810
Ad Hominem	0.742
Pathos	0.704

Table 4: Results of our ensembling model on the dev set, per-class.

suasive techniques to detect like 'Simplification', 'Distraction' and 'Reasoning' have lower performance.

5 Conclusion

In this paper, we describe the system team EURECOM used for sub-task 1 at SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. We explore multiple BERT-based models, training datasets, losses, data augmentation procedures, and training process. Our final submission consists of an ensembling model that performs majority voting between our top-3 best performing models for each persuasive technique. We find that some pre-trained models on harmful meme data are competitive, and that incorporating hierarchical information in the training process, such as outputting the whole 28 classes (including the ancestors) or using a hierarchical loss significantly improves the results. We obtain a hierarchical F1 score of 0.675 on the dev set and 0.655 (English), 0.345 (Bulgarian), 0.442 (North Macedonian), 0.177 (Arabic) on the test set.

Acknowledgements

This work has been partially supported by CHIST-ERA within the CIMPLE project (CHIST-ERA-19-XAI-003) and by ANR within the ECLADATTA project (ANR-22-CE23-0020).

References

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. In *14th International Workshop on Semantic Evaluation (SemEval)*, pages 1377–1414. International Committee for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. In *18th International Workshop on Semantic*

Evaluation (SemEval), SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 Task 6: Detection of Persuasion Techniques in Texts and Images. In *15th International Workshop on Semantic Evaluation (SemEval)*, pages 70–98. Association for Computational Linguistics.

Riba Edgar, Mishkin Dmytro, Ponsa Daniel, Rublee Ethan, and Gary Bradski. 2020. Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In *Winter Conference on Applications of Computer Vision*.

OpenAI et al. 2023. GPT-4 Technical Report.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2021. The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization.

Youri Peskine, Damir Korenčić, Ivan Grubisic, Paolo Papotti, Raphael Troncy, and Paolo Rosso. 2023. Definitions Matter: Guiding GPT for Multi-label Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4054–4063, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal Meme Dataset (MultiOFF) for Identifying Offensive Content in Image and Text. In *Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need.

A GPT-4-Turbo Prompts

For reproducibility, we share the exact prompt used to generate new examples using GPT-4-Turbo (as of January 2024):

```
[system] Your task is to generate short sentences that contains the <current_propaganda_technique> propaganda technique.
The definition of the <current_propaganda_technique> propaganda technique is the following: <current_propaganda_technique_definition>


Here are some examples:
 - <Random example x5>

([user] Please generate a short sentence that contains the <current_propaganda_technique> propaganda technique similar to the examples, on similar topics.
[assistant] <Random example>) x5
[user] Please generate a short sentence that contains the <current_propaganda_technique> propaganda technique similar to the examples, on similar topics.
```

TU Wien at SemEval-2024 Task 6: Unifying Model-Agnostic and Model-Aware Techniques for Hallucination Detection

Varvara Arzt, Mohammad Mahdi Azarbeik, Ilya Lasy, Tilman Kerl, and Gábor Recski

Faculty of Informatics, TU Wien
{varvara.arzt, mohammad.azarbeik, ilya.lasy, tilman.kerl, gabor.recski}@tuwien.ac.at

Abstract

This paper discusses challenges in Natural Language Generation (NLG), specifically addressing neural networks producing output that is fluent but incorrect, leading to "hallucinations". The SHROOM shared task involves Large Language Models in various tasks, and our methodology employs both model-agnostic and model-aware approaches for hallucination detection. The limited availability of labeled training data is addressed through automatic label generation strategies. Model-agnostic methods include word alignment and fine-tuning a BERT-based pretrained model, while model-aware methods leverage separate classifiers trained on LLMs' internal data (layer activations and attention values). Ensemble methods combine outputs through various techniques such as regression metamodels, voting, and probability fusion. Our best-performing systems achieved an accuracy of 80.6% on the model-aware track and 81.7% on the model-agnostic track, ranking 3rd and 8th among all systems, respectively.[1]

1 Introduction

In Natural Language Generation (NLG), the trade-off of prioritising fluency over accuracy results in neural networks generating "hallucinations" – outputs fluent but factually inaccurate. The automatic identification of such errors represents a substantial challenge (Huang et al., 2023; Ji et al., 2022). The SHROOM shared task on hallucination detection (Mickus et al., 2024) highlights concerns about the practical utility of fluently generated yet inconsistent outputs. In the SHROOM shared task, Large Language Models' (LLMs) outputs for definition modeling (DM), machine translation (MT), and paraphrase generation (PG) tasks are presented with input source text and corresponding 'gold' reference text. Notably, for PG, the input source text serves as the reference 'gold' text. While the training dataset lacks labels, an issue which is addressed in Section 3, the validation dataset includes binary labels of Hallucination or Not Hallucination and hallucination probability of 0 to 1, corresponding to Hallucination and Not Hallucination, respectively, for the LLM's output. These assessments, based on five annotators' evaluations, rely on determining if the model's output is supported by the 'gold' reference, from either the 'gold' target text, source text, or both, depending on the task (DM, MT, or PG).

The SHROOM dataset is categorised into two tracks: model-agnostic and model-aware. The model-aware track, in contrast to the model-agnostic, includes the specific LLM responsible for the provided output. This paper introduces methods tailored to both tracks and since the model-agnostic techniques can be applied to both, evaluations for these methods are conducted on both test datasets to provide a comprehensive analysis. The model-agnostic methods entail employing word alignment to establish semantic similarity between the model output and the 'gold' reference as well as fine-tuning a BERT-based model for hallucination detection. On the other hand, model-aware approaches delve into the analysis of hidden states and attention flow within the model architecture. Ultimately, a diverse set of ensemble techniques, comprising logistic regression with binary labels, linear regression with raw probabilities, voting, and probability fusion, are introduced to amalgamate the proposed methods.

2 Related Work

Since the tendency of LLMs to produce incorrect output poses a serious challenge in their application, the task of hallucination detection has recently attracted a variety of research work. Model-agnostic approaches include the training of dedicated machine learn-

[1]Our code is available at https://github.com/kleines-gespenst/shroom-hackathon

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1183–1196

ing models, such as a token-level classifier for detecting hallucination in machine translation (Zhou et al., 2021) or the BERT-based Vectara `hallucination_evaluation_model`[2], the latter which we also use in our model-agnostic experiments (see Section 3.2). Recent datasets for training and evaluating such models include task-agnostic corpora such as (Li et al., 2023) and HaDeS (Liu et al., 2022) as well as datasets focusing on specific generation tasks such as text summarisation (Laban et al., 2022), fact verification (Thorne et al., 2018), question answering (Pang et al., 2022; Longpre et al., 2021), or paraphrase generation (Zhang et al., 2019; Shen et al., 2022).

Another set of approaches involves comparing a model's output to some reference using any of a variety of unsupervised similarity metrics, including standard ngram-based measures such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) but also distributional similarity metrics such as BERTScore (Zhang et al., 2020), BARTScore (Yuan et al., 2021), or DiscoScore (Zhao et al., 2023). A large-scale analysis of the performance of these metrics on hallucination detection have been performed in the recent TRUE survey (Honovich et al., 2022). Further model-agnostic approaches to hallucination detection include comparing multiple LLM responses to a single query (Manakul et al., 2023), prompting LLMs to evaluate the likelihood of their own output being correct (Kadavath et al., 2022) as well as the use of external knowledge bases to assess the faithfulness of model outputs (Thorne et al., 2018; Guo et al., 2022; Peng et al., 2023)

Model-aware methods for hallucination detection include classifying an LLM's hidden layer activations to determine whether the question is answerable (Slobodkin et al., 2023) or whether its output is true (Azaria and Mitchell, 2023). The latter approach is the basis of the SAPLMA system, which we have also used in our experiments for the model-aware track of the shared task (see Section 3.3).

3 Methodology

Within the scope of hallucination detection, we employed both model-agnostic and model-aware methods. Our model-agnostic approaches encompassed rule-based techniques, featuring the application of string metrics and word embeddings, alongside the fine-tuning of pretrained language models. Model-aware methods leveraged the internal data of LLMs.

3.1 Automatic Data Annotation

Beyond the primary task of hallucination detection, a significant challenge arose due to the limited availability of labeled data, with only the SHROOM validation set being provided with labels. Faced with the absence of labeled training data, we explored diverse strategies for automatic label generation. These approaches encompassed zero-shot prompting with GPT-3.5 Turbo and the utilisation of BERTScore (Zhang et al., 2020) as a quantifiable metric, capable of serving as a probability indicator for hallucination within the generated content. BERTScore, employed in our approach for automatic data labeling, entails the aggregation of pairwise cosine similarity scores between the BERT contextual embeddings of tokens in candidate and reference sentences.

3.2 Model-agnostic Methods

Word alignment for semantic similarity Based on the understanding that hallucination is defined as model output that is semantically inconsistent with the reference output, we reduce the task of hallucination detection to that of measuring semantic similarity between pairs of sentences. The probability that a hypothesis sentence generated by a model contains hallucination should then be inversely proportional to its semantic similarity to the reference. In an effort to provide measures of semantic similarity that are more explainable than modern distributional metrics such as BERTScore (see Section 3.1) we developed a set of simple methods based on word alignment and word similarity. Given a word similarity metric, we simply align each word of one sentence to the most similar word in the other and define sentence similarity as the average of these similarities. Formally, for any word similarity metric S_w that maps any pair of words w_1, w_2 to the $[0, 1]$ range we define the similarity of sentences s_1 and s_2 as

$$S = \sum_{u \in s_1} \max_{v \in s_2} \frac{S_w(u, v)}{|s_1|} \quad (1)$$

We defined several word similarity metrics and for each determined a custom threshold t for which a hypothesis sentence s_{hyp} is considered a hallucination w.r.t. the reference s_{ref} if and only if

[2] https://huggingface.co/vectara/hallucination_evaluation_model

$1 - S(s_{hyp}, s_{ref}) \geq t$ (values of t were determined empirically on the validation dataset of the model-agnostic track). For each similarity type we experimented with alternative methods for calculating overall similarity from word similarities, including the harmonic mean and the minimum of the best similarities for each word, but the plain average (Eq. 1) yielded the best performance. We also ran all experiments with stopword removal using the `nltk` library (Bird et al., 2009) but found it to cause a slight decrease in performance. In our submissions, we included outputs based on two word similarity metrics. The `levenshtein` system that uses Levenshtein distance of word pairs as the word similarity measure for Equation 1, a string similarity metric defined as the number of character-level edit operations required to transform one word into another (Levenshtein, 1966). The similarity metric S_L was defined as $1/(1 + L)$ to obtain a value between 0 and 1 that is inversely proportional to the distance measure L. The `paragram` system combines the word alignment method with distributional similarity, here word similarity is defined as the cosine similarity of two words in the static English word embedding `paragram_300_SL999` (Wieting et al., 2015), which has been fine-tuned on the task of measuring word similarity on the SimLex-999 dataset (Hill et al., 2014).

Finetuning a BERT-based Hallucination Detection Model Another approach for the model-agnostic track encompassed finetuning a pretrained hallucination detection model based on BERT (Devlin et al., 2019). An open-source model developed by Vectara was chosen for that purpose as it achieved high accuracy on a range of hallucination detection benchmarks including e.g. accuracy of 76% on the `SummaC` dataset (Laban et al., 2022). Built upon the deberta-v3-base (He et al., 2021), Vectara undergoes initial training on Natural Language Inference (NLI) data, followed by subsequent fine-tuning on summarisation datasets. The model is trained utilising a cross-encoder architecture.[3] Given the scarcity of labeled data of high quality necessary for the initial finetuning of a language model, a departure from the approach taken by Zhou et al. (2021) was considered. In their work, they utilised XLM-R (Conneau et al., 2020) for hallucination detection within the scope of a machine translation task, and RoBERTa (Zhuang et al., 2021) for hallucination detection within the scope of summarisation task. However, due to the insufficient availability of labeled data, this method was not deemed applicable in the current study.

3.3 Model-aware Methods

Hidden States Model-aware techniques are based on analysing internal data of LLM during inference. One of the possible approaches is the analysis of the outputs of the hidden layers of the transformer. Using vector values of hidden layers for hallucination detection was proposed in a method called Statement Accuracy Prediction, based on Language Model Activations (SAPLMA) (Azaria and Mitchell, 2023). SAPLMA is a probing technique that utilises a feedforward neural network trained on activation values of the hidden layers of LLM.

Attention Flow We follow the attention-based token-level importance metric proposed by DeRose et al. (2020) for sequence classification and adopt it to sequence-to-sequence. We extract and analyse the attention weights from the model predictions and trace how the model shifts its focus from the output back to the input. This is done by summing and averaging the weights in the decoder, and then mapping these influences back through cross-attention layers to the encoder. Thereby, highlighting which parts in the input text are influential for the output. As an addition, we apply exponential decay to the influence scores to account for a diminishing impact of distant tokens across layers.

Consequently, we derive an influence matrix that quantifies the influence scores for each layer and token. Under the assumption that there exists a meaningful correlation between the input and its corresponding output, and thereby, also in cases where the output is characterised by hallucinations, these identified features can be leveraged to build a classifier.

4 Experiments

4.1 Automatic Data Annotation

As described in Section 3.1, two approaches were utilised for automatic annotation of the SHROOM training set, specifically zero-shot prompting with GPT-3.5[4] and the BERTScore metrics. To generate

[3] Notably, in contrast to the SHROOM dataset, Vectara produces a probability scale from 0 to 1, where 0 represents hallucination and 1 denotes factual consistency. Implementing a threshold of 0.5 enables predictions to assess the alignment of a document with its source.

[4] gpt-3.5-turbo-1106

probabilities with GPT-3.5 in a zero-shot manner, OpenAI API was used. The prompt was crafted specifically to incorporate multiple dataset samples in one pass in order to speed up the labeling process. Specifically, there were 32 samples in the prompt. Although the latest models support larger input context length, our experiments showed that passing more samples per request results in inconsistent and poor-quality labels. The input for the model was structured as pairs, consisting of a context ('tgt' for MT and DM, and 'src' for PG) and a sentence ('hyp'). Prompt engineering was inspired by the SHROOM baseline kit combined with instructions to return structured output in a JSON format. The explicit prompt passed as an instruction to GPT-3.5 is referenced in Appendix A. The total cost associated with utilizing the GPT-3.5 API amounted to ∼3$. Performance evaluation of GPT-3.5 on the validation dataset, comprising both its model-agnostic and model-aware parts, yielded metrics of **0.68** and **0.49** for accuracy and Spearman's correlation coefficient, respectively.

Simultaneously, BERTScore calculations were conducted on identical pairs of text instances used for probability generation with GPT-3.5. The candidate sentence ('hyp') and reference sentence ('tgt' for MT and DM, and 'src' for PG) served as inputs for BERTScore computation. The resulting BERTScore values, ranging between 0 and 1, were utilised in their raw form, wherein outputs of a specific LLM featuring BERTScores closer to 1 indicate a higher probability of being non-hallucinations. We utilised BERTScore version 0.3.13 with the RoBERTa Large model (Zhuang et al., 2021). Performance evaluation of BERTScore values on the validation dataset, comprising both its model-agnostic and model-aware parts, yielded metrics of **0.67** and **0.41** for accuracy and Spearman's correlation coefficient, respectively. For these calculations, the transformation of BERTScore values into labels utilized a threshold of 0.5.

4.2 Model-agnostic Methods

Word alignment The submissions `levenshtein` and `paragram` are based on the word alignment method described in Section 3.2. Threshold values for binary classification were determined empirically using the validation set of the model-agnostic dataset and set to 0.35 for `levenshtein` and 0.3 for `paragram`. Identical parameters were used to generate the submitted outputs for both model-agnostic and model-aware tracks of the shared task.

Finetuning a BERT-based Hallucination Detection Model This section presents the finetuning of Vectara pretrained hallucination detection cross-encoder model. We conducted finetuning on 4 different data combinations, including the validation set exclusively (`vectara-val`), the training set with probabilities generated by either GPT-3.5 (`vectara-gpt`) or BERTScore (`vectara-bertscore`), and a combination of both the validation set and the training set with GPT-3.5 probabilities (`vectara-gpt-val`). Given that Vectara predicts hallucination probability independently of a specific LLM, we concatenated both model-aware and model-agnostic datasets for additional finetuning to address limitations arising from their relatively small sizes. In addition to the mentioned data combinations, we performed separate finetuning on subsets of the validation set (`vectara-val-subset`), representing model-aware and model-agnostic validation sets. These finetuned models' predictions were later used in ensemble methods (Section 4.4). Despite acknowledging potential bias associated with finetuning using automatically generated probabilities, this approach was pursued due to the necessity of labeled data. To mitigate bias from GPT-3.5 and BERTScore probabilities, approaches (`vectara-val`, `vectara-gpt-val`, `vectara-val-subset`) utilised the validation set for finetuning. The finetuning process involved creating input instances for each training pair, comprising the LLM's generated text ('hyp'), the 'gold' reference text, and the probability of hallucination obtained from annotators ('p(Hallucination)'). The 'gold' reference text corresponds to the intended reference 'gold' text ('tgt') for MT and DM tasks, while for the PG task, where 'tgt' was mostly not provided in the SHROOM dataset, the model input ('src') served as the 'gold' reference text. Consistent hyperparameters were employed for finetuning vectara across all data combinations: the model was trained for 5 epochs with a batch size of 16 and a warmup of 0.1.

4.3 Model-aware Methods

Hidden States During our experiments with SAPLMA method (see Section 3.3), we used a feedforward neural network as an activations classifier. It features three hidden layers with decreasing numbers of hidden units (256, 128, 64), all util-

ising ReLU activations. We discovered that this approach requires more data than the validation set could provide. Therefore, the classifier was trained fully on the GPT-3.5 labeled training dataset. For each task, the dataset was fed into the original task model to get outputs of each hidden layer of the decoder for a final decoded token (EOS). Hallucination classifier was trained with binary cross entropy objective where inputs were original model layer activations and outputs were GPT labeled hallucination probabilities. The exact layer number is considered a hyperparameter in this case which was selected by grid search. Further experiments were focused on selecting the exact hidden layer of the original model that may contain most of the information regarding the uncertainty of the model. Based on evaluation on a validation set the best results are different for each task: layer #10 (out of 12) for MT, layer #1 (out of 12) for DM, layer #5 (out of 16) for PG.

Attention Flow For `att-flow`, we conducted our experiments using the `scikit-learn` library (Pedregosa et al., 2011). The feature matrices that we obtained, have dimensions $L \times T$ where L denotes the number of layers and T the number of tokens, can be quite large. To address this, we employed Principal Component Analysis for dimensionality reduction, achieving six components, chosen after a structural evaluation of component ranges across tasks. This dimensionality reduction significantly enhanced the classifier's performance. A Support Vector Machine with a Radial Basis Function kernel and Platt scaling was utilised for deriving probabilities and predictions. Out of alternative kernels, none yielded comparable results. Notably, the SVM was trained independently for each task, recognising that tasks may exhibit distinct attention flows, particularly on model-aware data.

4.4 Ensemble Methods

Finally, we also created simple ensemble models by combining the outputs of the individual systems presented in previous sections. We experimented with a variety of methods including simple voting, regression metamodels, and fusion of predicted probabilities.

Logistic regression The submission `mm-logreg` involved hallucination prediction with a logistic regression model trained on a small set of binary features that correspond to the labels predicted by individual systems. For the model-agnostic track we included labels from 5 systems, `levenshtein`, `paragram`, `vectara-gpt`, `vectara-bertscore`, and `vectara-val-subset`.[5] For the model-aware track, we also included the labels predicted by the SAPLMA method (Section 4.3). For each track, the model was trained on the respective validation dataset, using default settings of the `LogisticRegression` model in the `scikit-learn` library (Pedregosa et al., 2011).

Linear Regression with Raw Probabilities Submission `mm-linreg-probs` followed a similar methodology to `mm-logreg` with the distinction being utilisation of raw probabilities predicted by individual systems instead of labels and employment of linear regression instead of logistic regression. For both the model-agnostic and model-aware tracks we included probabilities predicted by the same systems utilised in `mm-logreg`. Like `mm-logreg`, for each track the model was trained on the respective validation dataset, using default settings of the `LinearRegression` model in the `scikit-learn` library (Pedregosa et al., 2011).

Voting Simple `voting` was implemented as an additional ensemble method, using the same set of systems as for the `mm-logreg` method. For each model output, we counted the number of systems that predicted the `Hallucination` label, the threshold for the number of votes required to make a positive prediction was a parameter of the system that we optimised on the validation sets. For both tracks, the optimal strategy was to require at least 2 votes (2 out of 5 for the model-agnostic track and 2 out of 6 for the model-aware track).

Probability Fusion The `prob-fusion` method is proposed as a weighted average fusion approach for combining predictions from multiple models in hallucination detection. Confidence scores for each model are determined as the squared absolute difference between the model's predicted probability and its neutral point, serving as weights in the fusion process. The final fused probability is obtained as the weighted sum of individual model probabilities:

[5]The metamodel considered probabilities from a Vectara model fine-tuned on either the model-aware or model-agnostic validation set, depending on the track.

Table 1: Accuracy (Acc) and Spearman's rank correlation (Corr) results for each of the proposed models on detecting hallucination on the test datasets.

Model	Agnostic		Aware	
	Acc	Corr	Acc	Corr
baseline	0.697	0.403	0.745	0.488
levenshtein	0.663	0.362	0.711	0.418
paragram	0.643	0.355	0.685	0.379
vectara-val	0.809	0.723	***0.806***	0.707
SAPLMA	-	-	0.593	0.137
att-flow	-	-	0.61	0.245
mm-logreg	0.801	0.665	0.801	0.636
mm-linreg	***0.817***	***0.737***	0.801	***0.712***
prob-fusion	0.793	0.673	0.783	0.654
voting	0.735	0.597	0.756	0.587

$$P_H = \sum_{i=1}^{N} \left(\frac{|P_i - NP_i|^2}{\sum_{j=1}^{N} |P_j - NP_j|^2} \times P_i \right), \quad (2)$$

where P_H denotes the fused probability of hallucination, and P_i and NP_i denote predicted hallucination probability and neutral point of model i, respectively.

5 Results

Table 1 provides insights into the accuracy and correlation metrics for each method concerning hallucination detection, spanning both model-aware and model-agnostic tracks compared to the `baseline` results introduced by the SHROOM organisers. Compared to the organisers' baseline, which employs the Mistral model in a zero-shot manner for hallucination detection, our best-performing systems are based either on finetuning methods (`vectara-val`) or ensemble approaches (`mm-linreg`). As detailed in Table 1, our leading model in the model-agnostic track, the `mm-linreg`, achieves an accuracy of 81.7% and a Spearman's rank correlation coefficient (ρ) of 0.737 and our leading model in the model-aware track, the `vectara-val`, achieves an accuracy of 80.6% and a Spearman's rank correlation coefficient (ρ) of 0.707 for predicting labels and estimating the probability of hallucination, respectively. According to the official ranking provided by the Helsinki NLP group[6], our team's results, TU Wien team, are ranked 3rd in the model-aware track among 46 participants and 8th in the model-agnostic track among 49 participants. As reported in Table 2 in the Appendix B, the finetuning of Vectara on the validation set improves its accuracy and correlation by 5.06% and 6.48% in the model-agnostic track and 1.5% and 2.3% in the model-aware track, respectively.

[6]https://helsinki-nlp.github.io/shroom/

6 Analysis

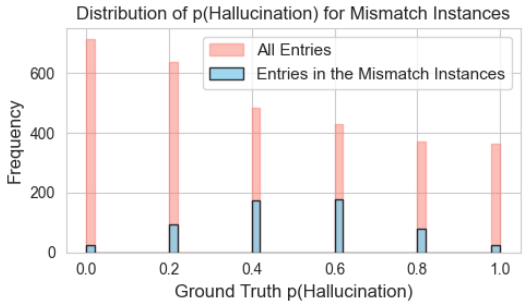


Figure 1: Distribution of hallucination probability for instances with disagreements between our top-performing `mm-linreg` predictions and the ground truth.

Figure 1 presents the distribution of ground truth p(Hallucination) entries in the entire test set encompassing both model-aware and model-agnostic test sets concatenated with those instances where the `mm-linreg` predictions mismatch the ground truth labels. The histogram highlights that the majority of p(Hallucination) entries across the entire ground truth test dataset are concentrated around 0. However, in instances where the metamodel predictions differ from the ground truth labels, there is a notable concentration around 0.4 and 0.6. This observation suggests that the model excels in predicting the correct label for non-controversial cases but encounters challenges in subjective scenarios, where the decision-making process becomes more intricate. Subsequent results, obtained by selectively filtering data based on p(Hallucination) values, reveal nuanced performance metrics: for instances where p(Hallucination) equals only 0 or 1, accuracy of 95.17% and Spearman Correlation (ρ) of 0.796 is achieved; when including p(Hallucination) values of 0, 0.2, 0.8, and 1, the accuracy remains high at 89.21% with a Spearman Correlation (ρ) of 0.731. In contrast, the analysis without filtering, yielding an accuracy of 80.87% and Spearman Correlation (ρ) of 0.724, underscores the impact of data filtering on the model's predictive perfor-

mance. This inherent subjectivity in hallucination detection, especially evident with a concentration around the controversial interval, underscores the complex nature of such judgments.

6.1 Qualitative error analysis

Building on the observations drawn from Figure 1, we conducted a more in-depth qualitative analysis of misclassifications within the entire test set. A subsequent in-depth manual analysis of misclassifications was carried out for each task independently. 46% of the misclassifications of the `mm-linreg` belong to the DM task, 33% of them belong to the MT, and 21% of them belong to the PG. For the MT task, we analysed 94 misclassified instances, covering all samples present in the test set for the Ru-En language pair[7], while for the DM and PG tasks, we analysed 20 randomly selected instances. While examining mislabeled samples, a crucial question arises about distinguishing between hallucinations and incorrect answers, such as inaccuracies in the DM task. In MT, ambiguity persists over categorising word-for-word translations, common among non-native speakers, as hallucinations or simply a translation that lacks accuracy. For instance, the model output *Her legs hurt* (MT instance with the id #826), as opposed to the 'gold' reference *Her feet ached*, was labeled by the annotators as a hallucination with a probability of 0.8. The Russian source text, У неё болели ноги, can be translated into English as both *Her feet ached* or *Her legs were hurting*, since the Russian ноги can refer to both *feet* and *legs*. Therefore, the only discrepancy in the model hypothesis lies in the incorrect grammatical form of the verb *hurt*. Analysing the aforementioned example and additional instances available in Appendix C suggests that, despite instructions for annotators to utilise either 'tgt' or 'src' as the 'gold' reference, annotators predominantly labeled the dataset using 'tgt' and 'hyp'. This discrepancy may pose a challenge. Additionally, the process of generating 'tgt' is only clarified for the DM task, with no explanation provided for the MT or PG tasks. The shared task documentation lacks explicit guidance on annotator instructions, stating only that annotators should verify if all information in the hypothesis is supported by the 'gold' reference. This formulation may lead to diverse interpretations of what constitutes a hallucination. Instances such as *If you persecute heretics or <define> discrepants </define> , they unite themselves as to a common defence [...]*[8] underscore the necessity for annotators to possess language proficiency, a requirement challenging to meet in a crowdsourced setting.

The error analysis of disagreements with 'gold' annotations highlights the subjective nature of hallucination detection, presenting a challenge for both machines and human annotators. The absence of a precise definition for hallucination further complicates the task. In NLG tasks like PG or DM, annotators require significant language proficiency or even a linguistic background. Instances with majority-based gold labels and low inter-annotator agreement (probabilities of 0.4 or 0.6) anticipate challenges for models, as these instances are ambiguous even for humans. Furthermore, qualitative analysis of the misclassifications suggests a tendency for annotators to mislabel longer texts. For instance, the model hypothesis for the MT instance with the phrase *The Beer of His Words Back*[9] corresponding to the gold reference *I stand corrected* was labeled by the annotators as a non-hallucination with a probability of 0.2. This may be attributed to the challenge of maintaining concentration when reading longer texts. This observation implies a heightened difficulty in detecting hallucinations in lengthier passages.[10] A comprehensive list of manually verified examples for all three tasks, accompanied by corresponding explanations, is extensively documented in Appendix C.

6.2 Complications with Model-aware Track

Our top-performing systems for both model-aware and model-agnostic test sets are based on model-agnostic approaches. However, the final results for our model-aware methods proved to be less promising, achieving about 60% accuracy on the test set (see Table 1). Possible reasons for the sub-optimal performance of the SAPLMA (4.3) method include the lack of reproducibility instructions for the model-aware track, poor quality of GPT-generated training labels, and the method's original design for decoder-only models, whereas all task models are encoder-decoder models. Results for the Attention Flow method (4.3) could also be enhanced through various techniques such as ad-

[7]The Ru-En language pair was selected because two authors of the paper are native Russian speakers.

[8]DM instance with the id #885 labeled by the annotators as a hallucination with a probability of 0.8

[9]Instance with the id #2251

[10]The average length of 'src' input in the SHROOM test set is 95 characters, or 17 tokens.

justing decay rates, incorporating decoder scores, and introducing feature weights.

During our experiments on model-aware datasets (train & validation set), we encountered challenges reproducing model outputs for two tasks: MT and DM. Using the provided Huggingface models with default parameters yielded different results than those shown in the dataset. Despite experimenting with numerous inference parameters from the Huggingface library, we could not obtain the same input-output pairs ('src'-'hyp') as in the dataset. This discrepancy significantly impacted label alignment, rendering samples labeled as hallucinations no longer hallucinations with newly generated results. This issue is crucial for the model-aware track, as SAPLMA and Attention Flow methods utilise internal data from forward pass for each sample but rely on labels from the dataset. We contend that this problem might substantially reduce the quality of these methods.

7 Conclusion

This paper outlines our systems for the SemEval shared task on LLM hallucination detection, covering both model-aware and model-agnostic subtasks. Our finetuned BERT-based model demonstrated strong performance, securing the 3rd rank in the model-aware track and underscoring the efficacy of our approach. Notably, our leading system in the model-agnostic track employs a metamodel that integrates predictions from diverse systems, including a finetuned BERT-based hallucination detection model, as well as rule-based methodologies and those relying on LLM hidden states.

The absence of a universally agreed-upon definition for hallucination complicates both human and machine evaluations, as evident in the error analysis of Section 6.1. Although attempts have been made to systematically define hallucination, such as by (Huang et al., 2023) and (Ji et al., 2022), the NLP community's understanding remains broad. This encompasses scenarios where a model outputs entirely false information or information close to the desired output but incomplete. Human annotators bring their world knowledge and views, influencing annotations and subsequently affecting model performance. Hallucination detection is challenging for both machines and humans, with achieving high inter-annotator agreement proving particularly difficult when the task definition is overly broad. Especially tasks like paraphrase generation or definition modeling, where numerous correct outputs are possible, are inherently subjective and tied to the annotator's real-world knowledge and beliefs (Heidegger, 2001; Honovich et al., 2022). A clearer definition of the annotation task, specifically detailing what constitutes hallucination for that task, could potentially enhance inter-annotator agreement and subsequently improve model performance. Detecting hallucinations across various NLP tasks poses a significant challenge. The SHROOM dataset encompasses three distinct tasks, whereas existing hallucination detection benchmarks often address only a single task.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

S. Bird, E. Klein, and E. Loper. 2009. *Natural language processing with Python*. O'Reilly Media.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Joseph F DeRose, Jiayao Wang, and Matthew Berger. 2020. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics*, 27:1160–1170.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*.

Martin Heidegger. 2001. On the Essence of Truth. In *The Nature of Truth: Classic and Contemporary Perspectives*. The MIT Press. _eprint: https://direct.mit.edu/book/chapter-pdf/2300694/9780262278690_car.pdf.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55:1 – 38.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics – doklady*, 10(8):707–710.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in

Python. *Journal of Machine Learning Research*, 12:2825–2830.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback.

Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3178–3190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625, Singapore. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Zhao, Michael Strube, and Steffen Eger. 2023. DiscoScore: Evaluating text generation with BERT and discourse coherence. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3865–3883, Dubrovnik, Croatia. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Prompt for GPT-3.5

```
You will be provided with a Sentence
and your task is to rate the
consistency of that sentence to
that of the provided Context. Your
answer must be only a number between
0.0 and 1.0 rounded to the nearest two
decimal places where 0.0 represents no
consistency and 1.0 represents perfect
consistency and similarity.
Reply with a valid JSON in following
format:
{"answers":{"<pair_id>": <float>}}.
Example:
{"answers":{"0":0.7,"12":0.33}}.
Array of answers should contain reply
for each Context/Sentence pair.
```

Listing 1: Dataset labeling prompt

B Vectara performance

The performance of the Vectara finetuned on different datasets is shown in Table 2.

Table 2: The performance (Accuracy (Acc) and Spearman's rank correlation (Corr) of the Vectara finetuned on different datasets.

Model	Agnostic		Aware	
	Acc	Corr	Acc	Corr
Standard vectara	0.770	0.679	0.794	0.691
vectara-gpt-val	0.745	0.69	0.776	0.695
vectara-bertscore	0.661	0.421	0.705	0.455
vectara-val	***0.809***	***0.723***	***0.806***	***0.707***
vectara-gpt	0.697	0.706	0.772	0.699

C Disagreement in Annotations

During the examination of misclassifications in MT, it was observed that approximately 15% (13 out of 94) of instances pertaining to the Ru-En language pair could be subject to different labeling by the authors of this paper. Furthermore, a detailed analysis of 20 randomly selected misclassifications from the PG and DM segments of the test set, revealed notable discrepancies. Specifically, 20% for the PG task (4 out of 20) and approximately 35% (7 out of 20) in the DM task would receive distinct labels according to the authors' assessment. Tables highlighting these disagreements for each of the three tasks are provided below for reference.

The MT example below was labeled by the annotators as a non-hallucination with a probability of 0.2 (instance with the id #2251 in the model-agnostic test set)[11]:

[11]The entire sample can be found in Table 4

src: Беру свои слова обратно.
tgt: I stand corrected.
hyp: The Beer of His Words Back.

Our top-performing `mm-linreg` labeled this instance as a hallucination with a probability of 0.54.

As previously deliberated, the categorisation of a translation lacking natural fluency as a hallucination is subject to scrutiny. This contention is exemplified by the following MT example (instance with the id #826 in the model-agnostic test set):

src: У неё болели ноги.
tgt: Her feet ached.
hyp: Her legs hurt.

This instance was labeled by the annotators as a hallucination with a probability of 0.8, while our `mm-linreg` labeled it as a non-hallucination with a probability of 0.2. In Russian, ноги stands for both *feet* and *legs*. The only discrepancy in the 'hyp' lies in the incorrect grammatical form of the verb *hurt*. An English native speaker would likely formulate a sentence such as *her legs were hurting*.

Upon scrutinizing misclassified instances in machine translation, discernible patterns emerged where our model encounters challenges. Specifically, it grapples with correctly identifying the presence of a suitable pronoun in the English translation (instance with the id #1791 in the model-agnostic test set):

src: Кажется, он сломан.
tgt: I think it's broken.
hyp: Looks like he's broken.

The form сломан represents a short form of the past participle of the perfective aspect of the verb сломать. This verb form is exclusively applicable when describing an inanimate object that has been broken. Notably, due to the gender agreement of Russian pronouns with the associated noun, the sentence includes the pronoun он, corresponding to the English pronoun *he*. To express the idea of an animate object being broken, as in the English phrase *he is broken*, the Russian equivalent would be он сломлен, where сломлен represents a short form of the past participle of the perfective aspect of the verb сломить. This sample was labeled by the annotators as a hallucination with a probability of 0.8. Our best-performing model labeled it as a non-hallucination with a probability of 0.2. It is pertinent to note that the annotation of such instances lacked consistency (instance with the id #69 in the model-agnostic test set):

src: На сколько врач тебе дал освобождение?
tgt: How long did the doctor sign you off school for?
hyp: How much did the doctor give you your release?

The annotators assigned a non-hallucination label to this instance with a probability of 0.4. Conversely, our best-performing model categorised it as a hallucination with a probability of 0.8. The English 'hyp' sentence appears somewhat unconventional, and alternatives such as *How long did the doctor grant you a release?* or *For how long did the doctor excuse you?* would convey a more natural phrasing. Notably, it is essential to acknowledge that the target ('tgt') reference, referred to as the 'gold' text, does not provide a fully accurate translation of the source Russian sentence. In the original Russian sentence, освобождение (translated as *release*) does not exclusively refer to a school release authorized by a doctor. Furthermore, it is pertinent to inquire about the methodology employed by the creators of the SHROOM dataset in generating the 'gold' text for the MT and PG sections, as this information is elucidated solely for the DM part of the dataset, indicating that the gold definition is sourced from Wiktionary.

It is imperative to acknowledge that our models were exclusively trained utilising 'tgt' and 'hyp' for both MT and DM, i.e. disregarding 'src'. Consequently, this means that our models cannot possess the capability to comprehend certain grammatical nuances of the Russian language, as the models were not trained on the Russian text. The decision to employ 'tgt' as a reference was motivated by the lack of statistical data regarding the languages encompassed in an MT part of the dataset since we could not use 'src' without specifying the language for a range of our approaches including finetuning of a BERT-based hallucination detection model. Considering the potential impact on system performance, an alternative approach could involve incorporating 'src' and 'hyp', or even 'src', 'tgt', and hyp' for the automatic generation of probabilities using GPT-3.5 for at least an MT part of the dataset. However, the quality of GPT-3.5 for low-resource languages cannot guarantee promising results for the plethora of languages encapsulated in the SHROOM dataset. Despite this caveat, such an approach may empower the model trained on such sort of automatically generated probabilities to better 'learn' the linguistic idiosyncrasies of a given language, which cannot be adequately captured solely through reliance on 'tgt'.

Below you can find an example for a PG task (PG instance with the id #2907 in the model-agnostic test set):

src: Does POTUS know?
tgt: Is the president aware of this?
hyp: Does POTUS know what you're doin' with?
gold_label: Not Hallucination
gold_prob: 0.2
label_metamodel: Hallucination
prob_metamodel: 0.78

A range of instances demonstrates that the SHROOM dataset annotation requires language proficiency from the annotators that cannot be expected in a crowdsourced setting[12] (DM instance with the id #885 in the model-agnostic test set):

src: If you persecute heretics or <define> discrepants </define> , they unite themselves as to a common defence [...]
tgt: (archaic) A dissident.
hyp: (obsolete) One who disagrees; a dissenter.
gold_label: Hallucination
gold_prob: 0.8
label_metamodel: Not Hallucination
prob_metamodel: 0.35

It is noteworthy to highlight that across all three tasks encompassed in the dataset, there exists a subset of instances that were correctly labeled but received a probability very proximate to 0.5. This scenario poses a potential challenge for models trained on such data (instance with the id #2998 in the model-aware test set):

src: The war with Serbia must be the last of Europe's civil wars.
tgt: -
hyp: Europe must end with the war with Serbia.
gold_label: Hallucination
gold_prob: 0.6
label_metamodel: Non-Hallucination
prob_metamodel: 0.4

[12] Organisers of the SHROOM shared task reported that the SHROOM validation dataset was built through crowdsourcing with 200 annotators involved annotating 20 instances each. Details about the test set annotations were not provided.

Table 3: PG Disagreement

id	Source	Target	Hypothesis	GT	GT p(H)	Pred	Pred p(H)
1150	I'll take the air.	I'm going out for some air.	I'm gonna take the air, mister.	Not Hallucination	0.40	Hallucination	0.58
2187	That was yesterday.	Well, it was a day ago now.	That was yesterday's news.	Not Hallucination	0.20	Hallucination	0.62
2907	Does POTUS know?	Is the president aware of this?	Does POTUS know what you're doin 'with?	Not Hallucination	0.20	Hallucination	0.79
674	Estimates of the number of displaced persons within Kosovo are now up to 850 000, according to NATO.	-	NATO says there are 850 000 displaced people in Kosovo.	Hallucination	0.60	Not Hallucination	0.14

Table 4: MT Disagreement

id	Source	Target	Hypothesis	GT	GT p(H)	Pred	Pred p(H)
55	Ты правда этого не знал?	Did you really not know that?	You don't know what?	Not Hallucination	0.40	Hallucination	0.58
69	На сколько врач тебе дал освобождение?	How long did the doctor sign you off school for?	How much did the doctor give you your release?	Not Hallucination	0.40	Hallucination	0.80
479	Никто не узнает.	No one will find out.	No one knows.	Not Hallucination	0.20	Hallucination	0.62
745	Я плохо спал вчера ночью, но хорошо спал ночью сегодня.	I didn't sleep well the night before last, but I slept well last night.	I slept badly last night, but slept well tonight.	Not Hallucination	0.40	Hallucination	0.83
836	Я лучше не буду отвечать на этот вопрос.	I'd rather not answer that question.	I better not answer that question.	Hallucination	0.60	Not Hallucination	0.17
846	Ты всё испортил!	You've ruined it!	You ruined everything!	Hallucination	1.00	Not Hallucination	0.47
1456	Тому пришла в голову блестящая идея.	Tom had a bright idea.	That's why a brilliant idea came to mind.	Not Hallucination	0.40	Hallucination	0.82
2136	Я жалею, что зря потратил на это своё время.	I regret wasting my time on that.	I regret the fact that I spent my time here.	Not Hallucination	0.20	Hallucination	0.67
2251	«Глупых вопросов не бывает». – «Как мог Леонардо Дикаприо изобрести Мону Лизу, если в XIX веке не было цвета?» – «Беру свои слова обратно».	There's no such thing as a stupid question. "How did Leonardo DiCaprio invent the Mona Lisa if there was no color in the 1800s?" "I stand corrected."	"There are no stupid questions." — "How could Leonardo Dicaprio discover Mona Lisa if there was no color in the 19th century?" — "The Beer of His Words Back."	Not Hallucination	0.20	Hallucination	0.54
2326	Она повторно вышла замуж, когда ей было за сорок.	She remarried when she was in her mid-forties.	She married again when she was 40.	Not Hallucination	0.40	Hallucination	0.72
2634	Как жизнь, Майк? - "Меня Том зовут".	How are you doing, Mike? "My name is Tom."	How's life, Mike? - "I'm Tom."	Hallucination	0.60	Not Hallucination	0.21

Continuation of Table 4							
id	Source	Target	Hypothesis	GT	GT p(H)	Pred	Pred p(H)
2642	Как думаешь, ты не мог бы внести десять долларов на подарок Тому ко дню рождения?	Do you think you could pitch in $10 for Tom's birthday present?	How do you think you wouldn't be able to bring ten dollars for a gift because of that birthday?	Not Hallucination	0.40	Hallucination	0.81
2727	Стоматологи рекомендуют менять зубную щётку каждые три месяца, потому что со временем её щетина всё хуже удаляет зубной налёт, а также в ней скапливаются микробы.	Dentists recommend to change toothbrushes every three months, because over time their bristles become worse at getting rid of plague, as well as accumulate microbes.	Dentists recommend changing the toothbrush every three months, because over time its bristle increasingly removes plaque, as well as microbes accumulate in it.	Not Hallucination	0.40	Hallucination	0.65

Table 5: DM Disagreement

id	Source	Target	Hypothesis	GT	GT p(H)	Pred	Pred p(H)
61	A grand distinction is to be drawn, in this respect, between the <define> swell mob </define> and common thieves; the former being, for the most part, men of the world, of some education — not appearing at all flash (thief - like), but, on the contrary, acting the part of gentlemen in society.	(archaic, slang) Well-dressed thieves and swindlers, regarded collectively.	(slang, dated) A group of thieves.	Not Hallucination	0.40	Hallucination	0.62
257	This is so because, as Kant already taught, the nonconsensual transfer of goods is only compatible with freedom when […]	In an omnilateral fashion.	In an omnidirectional manner.	Hallucination	0.60	Not Hallucination	0.14
885	If you persecute heretics or <define> discrepants </define>, they unite themselves as to a common defence […]	(archaic) A dissident.	(obsolete) One who disagrees; a dissenter.	Hallucination	0.80	Not Hallucination	0.35
266	Whilst the viewshed quantifies visibility for a limited set of test locations...	The view from a particular vantage point.	The area of a building or other structure that provides a view.	Not Hallucination	0.20	Hallucination	0.71
1405	Some areas were deluged with a month's worth of rain in 24 hours.	To flood with water.	To flood; to overwhelm.	Hallucination	0.60	Not Hallucination	0.44
1685	Through it, through what takes place, the celebrants try to obtain a result, to influence the course of the hoped for or dreaded events that either depend on the current dispositions of a divinity or […]	A person who officiates at a religious ceremony, especially a marriage or the Eucharist.	One who holds a ceremony.	Not Hallucination	0.40	Hallucination	0.53

silp_nlp at SemEval-2024 Task 1: Cross-lingual Knowledge Transfer for Mono-lingual Learning

Sumit Singh, Pankaj Kumar Goyal and **Uma Shanker Tiwary**
Indian Institute of Information Technology, Allahabad
{sumitrsch, pankajgoyal02003}@gmail.com
ust@iiita.ac.in

Abstract

Our team, silp_nlp, participated in all three tracks of SemEval2024 Task 1: Semantic Textual Relatedness (STR). We created systems for a total of 29 subtasks across all tracks: nine subtasks for track A, 10 subtasks for track B, and ten subtasks for track C. To make the most of our knowledge across all subtasks, we used transformer-based pre-trained models, which are known for their strong cross-lingual transferability. For track A, we trained our model in two stages. In the first stage, we focused on multi-lingual learning from all tracks. In the second stage, we fine-tuned the model for individual tracks. For track B, we used a unigram and bigram representation with suport vector regression (SVR) and eXtreme Gradient Boosting (XGBoost) regression. For track C, we again utilized cross-lingual transferability without the use of targeted subtask data. Our work highlights the fact that knowledge gained from all subtasks can be transferred to an individual subtask if the base language model has strong cross-lingual characteristics. Our system ranked first in the Indonesian subtask of Track B (C7) and in the top three for four other subtasks.

1 Introduction

The importance of semantic relatedness in language has been long recognized. Applications include sentence representation, question answering, and text summarization (Abdalla et al., 2023). Sentences can be related through either paraphrasal or entailment relations, or through broader commonalities such as shared topics, viewpoints, temporal origins, and logical connections.

This shared task (Ousidhoum et al., 2024b) aims to expand the scope of significant research in natural language processing (NLP) by incorporating 14 languages. The focus of the research is on semantic similarity and has predominantly been conducted in English. The languages included in the task are Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu.

The task requires predicting the degree of semantic textual relatedness (STR) between pairs of sentences in multiple languages. The task is to rank these sentence pairs based on their level of relatedness, with scores ranging from 0 (completely unrelated) to 1 (maximally related). This task is divided into the following three tracks.

Track A: Supervised Challenge was developing a system trained on labelled datasets provided for the 11 subtasks. Publicly available related datasets could be utilized, but no additional dataset could be used in this work.

Track B: Unsupervised The challenge was developing a system without using labelled datasets to measure semantic relatedness or similarity with the text units longer than two words only. We took advantage of unigram and bigram features with SVM regression and XGBoost.

Track C: Cross-Lingual The challenge was to develop a system without labelled datasets in the target language and with the use of labelled dataset(s) in at least one other language (subtask). All datasets of track A other than the targeted subtask are utilized for similarity prediction in this work.

Our system utilizes cross-lingual learning by implementing multi-stage training methods similar to those used in (Wang et al., 2022), (He et al., 2022), and (Singh and Tiwary, 2023) for track A. In the first stage, we selected pre-trained cross-lingual language models that cover the languages used in our task and fine-tuned them on a combined dataset of all subtasks in track A for five epochs. This created a model checkpoint that had knowledge of multiple languages relevant to our task. In the second stage, we fine-tuned the model checkpoint generated in the first stage for each track individually.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1197–1203

For track C, we created a dataset by combining all the data from track A, except for the targeted subtask, and fine-tuned it with the language model in a supervised manner. We used unigram and bigram bag-of-words representation with SVM-based regression (SVR) and XGBoost for each subtask in track B.

Our team achieved the best model for the Telugu and Marathi subtasks of track A by using the MuRIL large model (Khanuja et al., 2021). In the first stage, we fine-tuned the model for all three tracks (English, Telugu, and Marathi) and then fine-tuned the model checkpoints for Telugu and Marathi.

For all track A languages, we fine-tuned XLM-R for subtasks in the 1st stage, and we fine-tuned the checkpoint generated in stage one for all monolingual tracks in the 2nd stage.

For Track B, only monogram or bigram representation is allowed for supervised training. We obtained comparable results using both unigram and bigram representations in combination with SVR and XGBoost.

In Track C, we used all training data from Track A except for the current subtask data since the use of the same language data was not allowed in Track C. We adopted a cross-lingual transfer approach, where MuRIL gave the best result for the Hindi subtask, while XLM-R predicted the best results for the other subtasks.

The results for each subtask are presented in Table 2, 3 and 4, along with the baseline results. In addition to being multilingual, the key challenges of the task were the presence of many low-resource languages that lack proper pre-trained models for learning and the limited availability of training examples for some subtasks of Track A (see Table 1). To address these challenges, we utilized language models (LMs) with cross-lingual transferability, along with a two-stage training strategy. Our code can be found here[1].

2 Related Work

The Semantic Textual Similarity (STR) task 2015 (Agirre et al., 2015) had three subtasks. The findings showed that the UMBC PairingWords system achieved the best score by semantically differentiating distributionally similar terms (Han et al., 2015). In the subsequent STR task (Cer et al., 2017), there are seven tasks that concentrate on multilingual and cross-lingual pairs. Additionally, one sub-track will delve into MT quality estimation data. The team ECNU (Tian et al., 2017) achieved the highest score using ensembles of well-performing feature-engineered models with deep learning methods. These models used random forest (RF), gradient boosting (GB), and XGBoost (XGB) regression methods. However, statistical and machine learning models were not the best, as transformer-based models gained attention after (Devlin et al., 2019). These models are pre-trained on large amounts of data and fine-tuned for various downstream tasks. Researchers have created the sentence transformer (Reimers and Gurevych, 2019) architecture for finding similar sentences. It is useful when given multiple sentences corresponding to a sentence, and we need to find the most similar one. However, fine-tuning the sentence transformer with the downstream task requires proper alignment between the dataset on which the sentence transformer is pre-trained and the dataset of the downstream task. In this task, there are multiple subtasks associated with multiple languages. Therefore, motivated by the performance of cross-lingual transformer-based models, we have used transformer-based language models that have strong cross-lingual transferability. It has been seen that cross-lingual transferability has advantages in various NLP tasks (Singh and Tiwary, 2023; Wang et al., 2022).

3 Data

This shared task provided fourteen sets of monolingual data (Ousidhoum et al., 2024a). There were nine languages for track A, each with training, validation, and testing data. For tracks B and C, only validation and testing data were provided for the Afr, Arb, Hin, Ind, and Pan languages. The training, validation, and testing data distribution for all languages is tabulated in Table 1. The English language had over 5,500 training examples, while other languages had comparatively fewer data provided.

4 Methodology

Our system has chosen robust cross-lingual transfer models such as XLM-R (Conneau et al., 2020), which is pre-trained on over 100 languages, and MuRIL (Khanuja et al., 2021), which is pre-trained on all Indic and English languages, for track A and C. We have followed a two-stage training approach

[1] https://github.com/singhsumit1/Semeval-Semantic_textual-relatedness.git

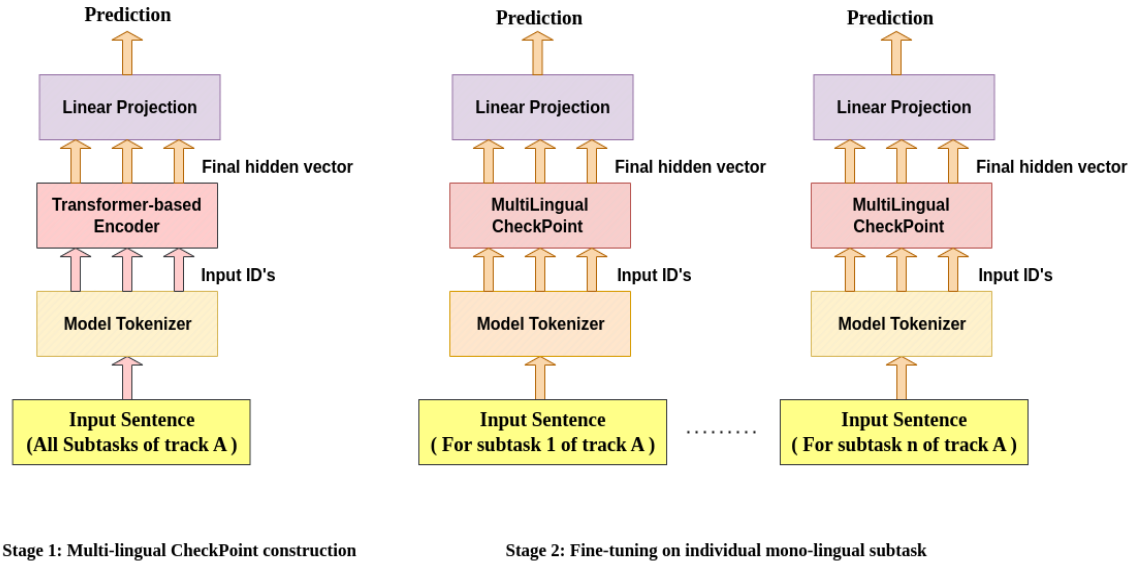


Figure 1: General Architecture of two-stage training.

Data	afr	amh	arb	arq	ary	eng	esp	hau	hin	ind	kin	mar	pan	tel
Train	-	992	-	1,262	925	5,500	1,562	1,763	-	-	778	1,155	-	1,146
Dev	375	95	32	92	70	250	140	212	288	144	102	293	242	130
Test	375	171	595	584	427	2,500	600	603	968	360	222	298	634	297
Total	750	1,258	627	1,938	1,422	8,250	2,299	2,578	1,256	504	1,102	1,746	876	1,573

Table 1: The datasets varied in the number of instances within their training, development, and test sets. Languages such as afr, arb, hin, ind, and pan lacked training data and were exclusively utilized in unsupervised and cross-lingual contexts.

for track A. The detailed methodology for each track is explained in the following subsections.

For tracks A and C, we combined both sentences of an example data with a special token of LM (</sep> for XLM-R and Roberta, and [SEP] for MuRIL) in the preprocessing stage. Model tokenizer tokenizes the combined sentence into tokens and generates token IDs and attention masks. For other subtasks, we utilized XLM-R and MuRIL with the two-stage training approach.

4.1 Methodology for track A

4.1.1 Two-stage training

In the initial stage of our project, we conducted multi-lingual training by utilizing the training data of all subtasks of track A in a selected LM that supports them. As shown in Fig. 1, our model used the annotations of all subtasks of track A in this stage. We performed experiments using various hyperparameters across five epochs, selecting the best multilingual checkpoint based on the average validation data loss. In the second stage, we fine-tuned the multilingual checkpoint from the first stage and utilized it as an initial model for fine-tuning each monolingual subtask in track A. We trained each mono-lingual track with different hyper-parameters in the second stage and selected the final model based on the validation data loss of the corresponding subtask.

4.1.2 Model Architecture

Figure 1 shows that the model tokenizer first tokenizes the input sentence. To improve GPU utilization, the tokenizers are set to a length of 92. Next, the language model generates word embeddings for each token. The embedding of the first token (which is <s> for the XLM-R and [SEP] for MuRIL) is passed through a linear layer, which projects it into logits, a vector of size one that represents the predicted similarity. Finally, we apply the Mean Square Error (MSE) loss function to calculate the difference between the prediction and the ground truth.

4.2 Methodology for track B

For track B, the sentences were converted into unigram and bigram representation and Support Vec-

Language	A1 arq	A2 amh	A3 eng	A4 hau	A5 kin	A6 mar	A7 ary	A8 esp	A9 tel
XLM-R (one-stage)	0.49	0.49	0.78	0.68	0.23	0.86	0.63	0.58	0.78
MuRIL (one-stage)	-	-	0.77	-	-	0.856	-	-	0.838
XLM-R (two-stage)	0.59	0.84	0.84	0.72	0.49	0861	0.81	0.66	0.789
MuRIL (two-stage)	-	-	-	-	-	0.862	-	-	0.842
Baseline_Score	0.6	0.85	0.83	0.69	0.72	0.88	0.77	0.7	0.82
Rank	7	8	6	5	15	13	10	17	6

Table 2: Results of all subtasks of track A has been tabulated for both the settings one-stage and two-sage with both the LMs XLM-R and MuRIL.

tor Regression[2] (SVR) (Fu et al., 2016; Liu et al., 2017).

4.2.1 Unigrams /Bigrams embeddings

Both sentences in the examples are combined and transformed into a vector. To create the vector, each sentence is indexed based on the presence of unigrams/bigrams, and the corresponding index value is filled with the count of unigrams/bigrams. The resulting vector is then fed into the SVR model along with the label values for training.

4.3 Methodology for track C

The methodology followed in track C is similar to the first stage of track A, with the difference that the combined dataset includes all subtasks except for the targeted subtask. For instance, to build a system for the English (eng) subtask, all data from the subtasks of track A, except for the eng subtask, is collected. The model architecture is also similar to that of track A but with only one stage involved.

5 Experimental setup

5.1 Track A and Track C

We achieved our best score using the MuRIL setup for the Telugu and Marathi subtasks, while the XLM-R setup worked best for the other track. During the training process, we experimented with different learning rates (5e-6, 2e-5, 5e-5, 8e-5, and 1e-4) and batch sizes (16, 32, and 64) in both stages. We selected the best model based on validation loss after five epochs of training.

For track A and track C, we used the setups outlined in Fig. 1. We implemented our task using the xxxTokenClassification class defined in (Wolf et al., 2020) for regression, where xxx refers to the selected model. We set the number of labels to one. The other hyperparameters for achieving the best results with both language models are listed in Table 5.

[2]Support Vector Regression

5.2 Track B

For subtasks which are training data available in track, we have generated monogram and bigram embedding and performed supervised learning with support vector regression (SVR) and gradient Boosting regression (XGBoost) with the Scikitlearn[3] library.

Evaluation metrics Results are Pearson correlation coefficient, which shows the similarity between two sentences.

6 Results and Analysis

The table below shows the Pearson score with official rank for Track A, Track B and Track C, along with the baseline score. Please refer to Table 2, 3 and 4 for more details. Our system performed exceptionally well in the Indonesian (ind) subtask of track B (B7), achieving 1st rank with a Pearson score of 0.53%. We secured 3rd rank in the three subtasks: B15, B10 and C10.

Track A: Two-stage MuRIL setup achieved the best scores for Telugu and Marathi subtasks, while two-stage XLM-R setup achieved the best score for all other subtasks.

Comparison between one-stage and two-stage methods with XLM-R LM. A comparison has been illustrated in Fig. 2. Based on the average performance of all subtasks in track A, it can be inferred that the two-stage strategy outperforms the one-stage strategy. The average score for all subtasks using the two-stage strategy was 0.73, while the average score for all subtasks using the one-stage strategy was 0.61.

Comparison between MuRIL and XLM-R for the Telugu and Marathi Table 2 shows that for Telugu and Marathi, MuRIL performed better than XLM-R. Two-stage MuRIL produces a 0.05 higher score for Telugu subtask than two-stage XLM-R. For Marathi, the Two-stage MuRIL produces slightly more than the Two-stage XLM-R.

[3]scikit-learn.org/stable

Language	B1 afr	B2 arq	B3 amh	B4 eng	B5 hau	B6 hin	B7 ind	B8 kin	B9 arb	B10 ary
Unigram SVR	-	0.4	0.31	0.39	0.41	-	-	0.47	-	0.68
Bigram SVR	-	0.4	0.64	0.32	0.39	-	-	0.35	-	0.55
Unigram XGBoost	-	0.3	0.4	0.28	0.35	-	-	0.38	-	0.72
Bigram XGBoost	-	0.31	0.4	0.33	0.33	-	-	0.37	-	0.72
Dice Loss	0.73	0.44	0.69	0.74	0.42	0.57	0.53	0.36	0.31	0.6
Baseline_Score	0.74	0.43	0.72	0.68	0.16	0.93	0.68	0.74	0.56	0.27
Rank	5	4	5	10	3	6	1	6	6	3

Table 3: All the results for subtasks of Track B have been displayed. For subtasks B1, B7, B8, and B9, labelled data was not provided, so only the Pearson scores predicted by the Dice loss are shown. For the other subtasks, the Pearson scores are displayed for unigram SVR, bigram SVR, unigram XGBoost, bigram XGBoost, and dice loss.

Language	C1 afr	C2 arq	C3 amh	C4 eng	C5 hau	C6 hin	C7 ind	C9 arb	C10 ary	C12 esp
Cross-lingual (XLM-R)	0.7468	0.3867	0.8048	0.7372	0.6428	0.7476	0.4716	0.4267	0.6732	0.5691
Cross-lingual (MuRIL)	-	-	-	-	-	0.8008	-	-	-	-
Baseline_Score	0.79	0.46	0.84	0.8	0.64	0.76	0.47	0.61	0.4	0.62
Rank	7	6	5	6	4	5	5	6	3	9

Table 4: Table shows the results of all subtasks of track C. MuRIL LM support Hindi (C6) subtask of the track C therefore only Pearson score of C6 given for the MuRIL LM.

track B Pearson score of track B tabulated in Table 3. It is clear from Table 3 that SVR performed better than XGBoost. The performance of SVR with unigram and bigram is not straightforward. Results showed that for B4, B5, B8, and B10, bigram embeddings perform better than unigram embeddings. However for the B3 unigram performed better.

track C Pearson score of the track C also showed in Table 4. Table 4 shows that for Hindi (C6), MuRIL performed better than XLM-R. All the other subtasks of this track are only predicted by the XLM-R in cross-lingual settings.

7 Conclusion

In this paper, we utilized multi-lingual track knowledge for the STR shared task to enhance the performance of monolingual models. Our team achieved first rank in the B7 subtask and third rank in the B5, B10, and C10 subtasks. We demonstrate that two-stage fine-tuning can help the monolingual models learn from the training data of all languages, leading to better performance. The results of track C illustrate the effectiveness of cross-lingual learning in a zero-shot scenario.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Cheng Fu, Bo An, Xianpei Han, and Le Sun. 2016. ISCAS_NLP at SemEval-2016 task 1: Sentence similarity based on support vector regression using multiple features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 645–649, San Diego, California. Association for Computational Linguistics.

Lushan Han, Justin Martineau, Doreen Cheng, and Christopher Thomas. 2015. Samsung: Align-and-differentiate approach to semantic textual similarity. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 172–177, Denver, Colorado. Association for Computational Linguistics.

Jianglong He, Akshay Uppal, Mamatha N, Shiv Vignesh, Deepak Kumar, and Aditya Kumar Sarda. 2022. Infrrd.ai at SemEval-2022 task 11: A system for named entity recognition using data augmentation, transformer-based sequence labeling model, and EnsembleCRF. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1501–1510, Seattle, United States. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages.

Wenjie Liu, Chengjie Sun, Lei Lin, and Bingquan Liu. 2017. ITNLP-AiKF at SemEval-2017 task 1: Rich features based SVR for semantic textual similarity computing. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 159–163, Vancouver, Canada. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sumit Singh and Uma Tiwary. 2023. Silp_nlp at SemEval-2023 task 2: Cross-lingual knowledge transfer for mono-lingual learning. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1183–1189, Toronto, Canada. Association for Computational Linguistics.

Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 task 1: Leverage kernel-based traditional NLP features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 191–197, Vancouver, Canada. Association for Computational Linguistics.

Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yueting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. 2022. DAMO-NLP at SemEval-2022 task 11: A knowledge-based system for multilingual named entity recognition. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1457–1468, Seattle, United States. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

8 Appendix

8.1 Details of Hyper-parameters

Table 5 shows the details of Hyper-parameters of best models for MuRIL and XLM-R setups with two-stage setup for Track A.

8.2 Comparison of Results of Track A

A comparison of subtasks of track A with one-stage XLM-R and two-stage XLM-R are shown in Fig. 2. For all the subtasks two-stage architecture performed better than one-stage architecture.

Hyper parameters	MuRIL setup	XLM-R setup
Baseline language model for first stage	google/MuRIL-large-cased	XLM-Roberta-large
Loss function	MSE	MSE
Hidden size for language model	1024	1024
Learning rate for language models	5e-05	5e-05
First-stage training epochs	5	5
Second-stage training epochs	5	5
Batch size	64	64
Dropout rate	0.1	0.1
Optimizer	AdamW	AdamW

Table 5: Hyper-parameters for MuRIL and XLM-R setups with two-stage setup for Track A.

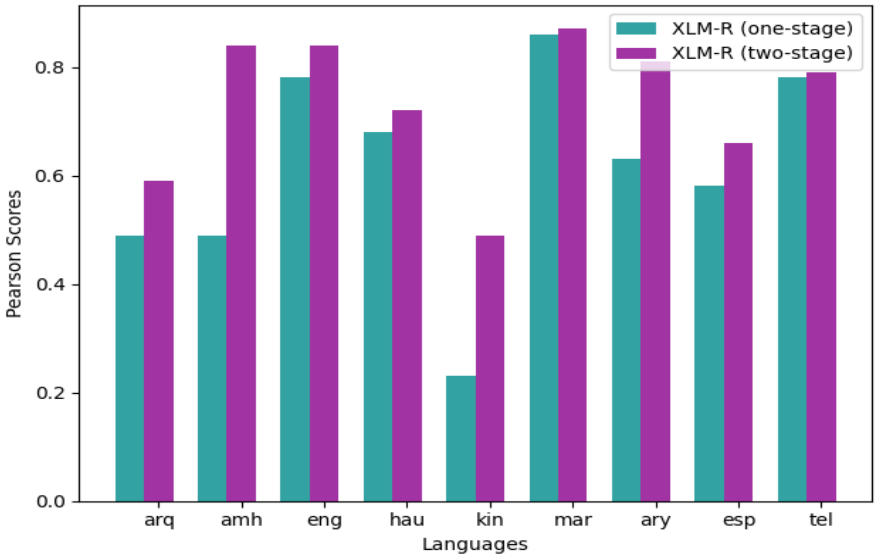


Figure 2: A comparison of subtasks of track A with one-stage XLM-R and two-stage XLM-R.

LastResort at SemEval-2024 Task 3: Exploring Multimodal Emotion Cause Pair Extraction as Sequence Labelling Task

Suyash Vardhan Mathur*
IIIT Hyderabad
suyash.mathur@research.iiit.ac.in

Akshett Rai Jindal*
IIIT Hyderabad
akshett.jindal@research.iiit.ac.in

Hardik Mittal
IIIT Hyderabad
hardik.mittal@research.iiit.ac.in

Manish Shrivastava
IIIT Hyderabad
m.shrivastava@iiit.ac.in

Abstract

Conversation is the most natural form of human communication, where each utterance can range over a variety of possible emotions. While significant work has been done towards the detection of emotions in text, relatively little work has been done towards finding the cause of the said emotions, especially in multimodal settings. SemEval 2024 introduces the task of Multimodal Emotion Cause Analysis in Conversations, which aims to extract emotions reflected in individual utterances in a conversation involving multiple modalities (textual, audio, and visual modalities) along with the corresponding utterances that were the cause for the emotion. In this paper, we propose models that tackle this task as an utterance labeling and a sequence labeling problem and perform a comparative study of these models, involving baselines using different encoders, using BiLSTM for adding contextual information of the conversation, and finally adding a CRF layer to try to model the inter-dependencies between adjacent utterances more effectively. In the official leaderboard for the task, our architecture was ranked 8th, achieving an F1-score of 0.1759 on the leaderboard. We also release our code here[1].

1 Introduction

Emotion Analysis is one of the fundamental and earliest sub-fields of NLP that focus on identifying and categorizing emotions that are expressed in text. Earlier, research in this domain focused on Emotion Detection in news articles and headlines (Lei et al., 2014; Abdul-Mageed and Ungar, 2017). However, later Emotion Recognition in Conversation gained popularity due to the widespread availability of public conversation data (Gupta et al., 2017). Recently, the task of emotion cause analysis has gained traction, which tries to identify the causes behind certain emotions (Xia and Ding, 2019a). This has widespread application such as building chatbots that can identify the emotions of the user and even identify the cause behind the emotions to perform certain actions (Pamungkas, 2019). For instance, companies can identify causes behind dissatisfaction in customer interactions and take appropriate measures (Yun and Park, 2022), AI-driven therapeutic insights can be gained using such models (DAlfonso, 2020), social media content moderation can be better done (Sawhney et al., 2021), work management and team management by companies can be improved (Benke et al., 2020).

In the task Wang et al., 2024, we tackle the problem of Multimodal Emotion Cause Pair Extraction, where given a set of utterances in a conversation, we must identify the following:

1. Emotion of every utterance (if any). These emotions can be one of Ekman's six basic emotions (Ekman et al., 1999).

2. Cause of these emotions, which is considered as the utterance that explicitly expresses an event or argument that is highly linked to the corresponding emotion.

Our proposed system tackles the task in a 3-step fashion – (a) First, we train a model to identify the emotions that are expressed in individual utterances in a conversation. (b) Next, we train a model to identify whether an utterance can be a cause of an emotion expressed in another/same utterance (candidate causes). (c) Finally, we train a model to pair emotion-utterances with their causes among the possible candidate causes. For both the (a) and (b) models we experiment with 3 basic architectures – (i) a simple Neural Network to determine the class of emotion (N-class classifier) and another Neural Network to identify whether the utterance is a candidate cause or not (binary classification). (ii) A BiLSTM (Sak et al., 2014) architecture that accounts for the surrounding context

*Equal contribution.

[1] github.com/akshettrj/semeval2024_task03

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1204–1211

of the conversation while doing the N-class and binary-classification. (iii) A BiLSTM CRF (Lafferty et al., 2001) architecture which accounts for the surrounding emotions as well while doing the N-class classification. We also experiment with different encoders for the three modalities.

2 Background

2.1 Dataset

The dataset used for this problem is **Emotion-Cause-in-Friends** prepared by Wang et al., 2023 specifically for this task. It has been prepared using conversations from the popular 1994 sitcom *Friends* as the source. This dataset contains 1,344 conversations made up of a total of 13,509 utterances, each conversation containing an average of 10 utterances. For each utterance, the dataset has an annotated transcript (covering text modality) and the corresponding video clip (covering visual and auditory modalities) from the show.

Each utterance is annotated with the emotion depicted by it, which is one of: `anger`, `disgust`, `fear`, `joy`, `neutral`, `sadness` and `surprise`. The dataset is highly skewed in terms of the frequency of different emotions in the dataset (see Figure 2). Further, the emotion-causes pairs for all the non-`neutral` utterances are provided in the dataset in a separate list.

The task MC-ECPE expects the model to take a list of such conversations and predict the emotion and emotion-cause pairs labels.

2.2 Related Work

A lot of work has been done in the field of emotion analysis in textual settings. Soon, work began on extracting not only the emotion but also the cause of that extracted emotion. People employed mainly two approaches for emotion cause analysis - 1. Extracting the potential causes given an emotion (Lee et al., 2010; Chen et al., 2010; Gui et al., 2016b) and 2. Extracting the emotion-cause pairs jointly (Xia and Ding, 2019b; Ding et al., 2020; Wei et al., 2020).

Poria et al., 2020 was the first to introduce the task of extracting emotion-cause in conversations but their focus was also only on the textual dialogues. However, in our natural way of conversation, we rely on things like facial expressions, voice intonations for determining the emotion of the speaker. We also rely on auditory and visual scenes to determine the cause of the speaker's emotions. Hence, it is clear that **Emotion-Cause Pair Extraction (ECPE)** is a multimodal task requiring at least three modalities: **text, audio** and **video**. Busso et al., 2008; McKeown et al., 2012; Li et al., 2022a and Poria et al., 2019 worked in the field of multimodal emotion analysis in conversations but they did not consider the emotion causes.

The task of MC-ECPE was first worked on by Wang et al., 2024.

3 System Overview

3.1 Baseline I: Utterance labeling

Our baseline model treats the problem as a simple **utterance labeling task**. We use pre-trained text, audio, and image encoders to encode the individual modalities and use these to train three models that can identify the emotions in the utterances, the candidate cause utterances, and finally identify valid emotion and cause utterance(s) pairs.

- **Text Encoding:** For encoding the transcription of each utterance, we use pre-trained BERT (Devlin et al., 2018) embeddings as the baseline embeddings. Additionally, we finetune DeBERTa-Base (He et al., 2020) on the training data for our experiments. DeBERTa makes use of a disentangled attention mechanism and an enhanced masked encoder to improve upon BERT's performance in a variety of tasks. Finally, we also tried RoBERTa-Large and (Liu et al., 2019) pre-trained EmotionRoBERTa-Base[2] which is publicly available RoBERTa-base model finetuned on the Go Emotions dataset (Demszky et al., 2020). For every text encoder, we perform mean-pooling of the word embeddings to get the textual representation of the utterance.

- **Video Encodings:** For encoding the videos, we sampled 16 equally spaced frames from the video and mean-pooled the embeddings for the 16 frames. For encoding these 16 images, we used MViTv2-small (Li et al., 2022b) encoder, which achieves state-of-the-art performance on the Kinetics video detection task (Kay et al., 2017), which makes it an obvious choice for recognizing activities happening in the conversations relevant for emotion/cause detection.

[2] https://huggingface.co/SamLowe/roberta-base-go_emotions

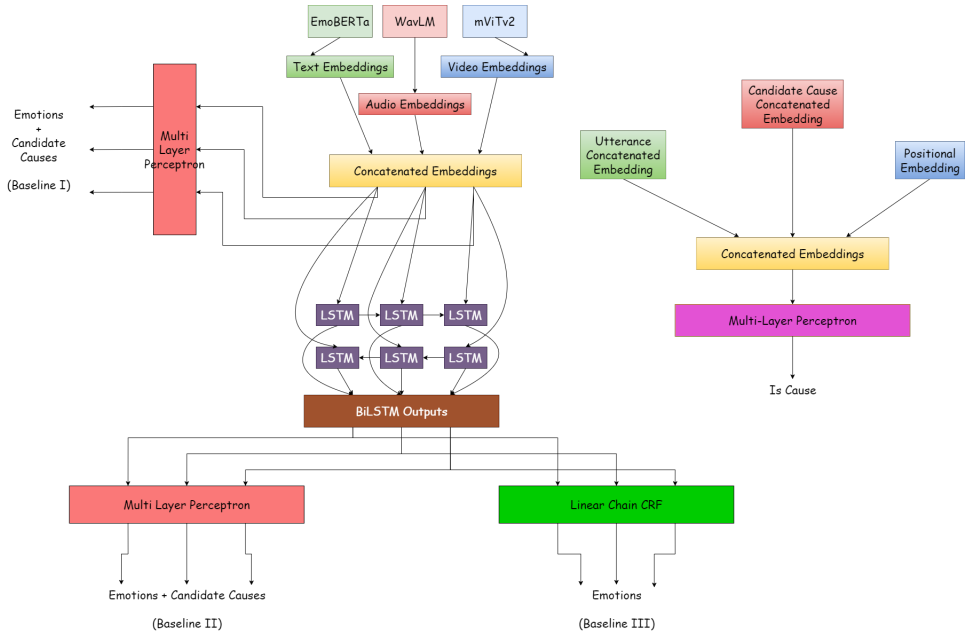


Figure 1: Model Architecture

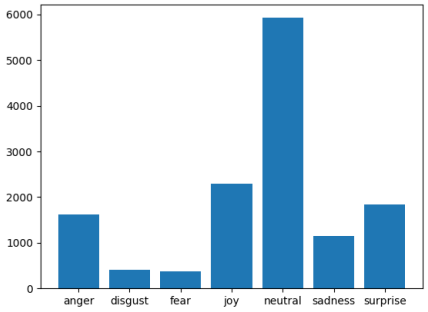


Figure 2: Emotion frequency in the dataset

- **Audio Encodings:** We used WavLM (Chen et al., 2022) for generating audio embeddings, which is trained on large audio data using masked speech representation and denoising in pre-training, making it suitable for various downstream speech tasks. We also try Wav2Vec2-Large (Baevski et al., 2020), which is trained by masking speech input in latent space and solving a contrastive task defined over a quantization of the latent representations which are jointly learned.

The model architecture is a combination of **three steps**, each of which is described below:

Step 1 – Emotion Classification

First, we concatenate the text, audio and video embeddings from the respective encoders and pass these concatenated embeddings into a dense layer, on which a Softmax function is applied to get the probability distribution over 7 classes (6 emotions and one neutral class). Due to a skewed distribution of the emotion labels in the dataset, we make use of **weighted Cross Entropy loss** to train the model, where the weights are taken as inverse of the frequency of the labels in the training dataset.

Step 2 – Candidate Cause Identification

For identifying the candidate cause, we similarly pass concatenated embeddings through a dense layer with a Sigmoid function, which predicts the probability of whether the utterance is a candidate cause or not. Binary Cross Entropy Loss is used to train the model.

Step 3 – Emotion-Cause pairing

For pairing the emotion utterances with the candidate causes, we concatenate the represenations for the emotion utterance and the cause utterance, with a distance embedding. This distance embedding is generated by giving positional embedding to each utterance, sampled from a Normal Distribution. This representation is passed through a dense layer with a Sigmoid function, which learns to predict the probability of the emotion-cause utterance pair being a valid emotion-cause pair or not for the given conversation, trained using Binary Cross Entropy Loss.

3.2 Baseline II: BiLSTM Architecture

The BiLSTM architecture is inspired by the work in Wang et al., 2024. While the Baseline I architecture treats the emotion and cause classification independently for each utterance, it is dependent on the surrounding context of the conversation too. Thus, the BiLSTM architecture models the problem as a **Sequence Labeling task**. We use the best encoders in the Baseline I architecture for generating the embeddings in this architecture.

Step 1 – Emotion Classification

Similar to the Baseline Model, we concatenate the embeddings of the three modalities, and pass them to a stacked BiLSTM. On top of the BiLSTM outputs, we apply a 7-class classifier to obtain the emotion category distribution. Similar to the Baseline I, weighted cross-entropy loss is used.

Step 2 – Candidate Cause Identification

For Candidate Cause prediction, similarly, the concatenated embeddings are passed through a BiLSTM on top of which a binary classifier is applied.

Step 3 – Emotion-Cause Pairing

The Emotion-Cause pairing model remains the same in this architecture as the Baseline I model.

In this architecture, BiLSTM provides the advantages of bidirectional and longer contexts which should help understand the emotions present in utterances better. This is because in a conversation, it is possible that the emotions are not just dependent on the current utterance, but on surrounding multimodal utterances as well.

3.3 Baseline III: BiLSTM-CRF Architecture

In the BiLSTM model, each classification decision was conditionally independent. Linear-chain CRFs are models generally used to model structured data where one output influences its neighboring outputs as it models the various transition probabilities, and have been extensively used with BiLSTMs for sequence labeling (Huang et al., 2015). This could be useful for emotion predictions because the emotion of one utterance is generally influenced by the emotions in its previous utterances. For instance, an utterance with happiness generally tends to be followed by another happiness utterance.

Step 1 – Emotion Classification

For this architecture, we add a CRF layer on top of the BiLSTM layers, and make use of the CRF-loss to train the model instead of Cross-Entropy loss as in the previous architectures. This loss models the transitions between the labels in the architecture, modelling the task as a more complex sequence labeling task. Thus, while the BiLSTM layer learns more about the language and emotions expressed through the language, the CRF layer tries to learn about the relations between the emotions.

Step 2 – Candidate Cause Identification

For Candidate Cause prediction, the architecture remains the same as in Baseline II. This is because the transitions between cause labels (being cause of an emotion in an utterance or not) does not make intuitive sense, and using BiLSTMs to capture surrounding context from other utterances is what seems more appropriate.

Step 3 – Emotion-Cause Pairing

The Emotion-Cause pairing model remains the same in this architecture as the Baseline I & II models.

4 Experimental Setup

We perform a random shuffle and use a 90-10% split for the train-validation split. The test set was provided by the authors, but its gold labels have not been made public.

The experiments involving Baseline II and III use *EmotionRoBERTa + WavLM + MViTv2* configuration. All the experiments involve applying a dropout of 0.3 on the audio, visual and textual embeddings before they are passed on to the main architectures. The BiLSTM for emotion detection consists of 4 layers while the one for candidate cause identification contains 3 layers. The dropout between the stacked layers of the BiLSTM is kept 0.3 as well. We use AdamW optimizer for all the three models, and use a linear learning rate scheduler with warmup for training the models. The Emotion Classification model is trained for 60 epochs, the Candidate Cause Identification model is trained for 40 epochs, and the Emotion-Cause Pairing Model is trained for 40 epochs as well.

In order to train the Emotion-Cause pairing model, we create positive and negative pairs during training. However, while the number of positive pairs is of the order N, the number of negative

Model Name	Emotion Detection			Candidate Cause Detection			Emotion-Cause Pairing			Emotion-Cause Pairing (Eval.)		Leaderboard	
	P	R	F1	P	R	F1	P	R	F1	wt. F1	Macro F1	wt. F1	Macro F1
Baseline I													
BERT + WavLM + MViTv2	0.61	0.52	**0.55**	0.71	0.71	**0.71**	0.93	0.87	**0.89**	**0.26**	0.20	0.182	0.165
EmotionRoBERTa + WavLM + MViTv2	0.55	0.45	0.47	0.67	0.66	0.66	0.93	0.86	**0.89**	0.20	0.18	**0.187**	0.170
DeBERTa (finet.) + WavLM + MViTv2	0.44	0.36	0.38	0.60	0.60	0.60	0.92	0.85	0.87	0.10	0.10	0.094	0.094
RoBERTa-L + WavLM + MViTv2	0.59	0.47	0.49	0.66	0.65	0.66	0.93	0.86	0.88	0.21	0.19	0.180	0.165
EmotionRoBERTa + Wav2Vec2 + MViTv2	0.55	0.47	0.48	0.67	0.67	0.67	0.93	0.87	**0.89**	0.21	0.20	0.172	0.170
BiLSTM (**Baseline II**)	0.55	0.51	0.52	0.67	0.67	0.67	0.93	0.86	**0.89**	0.22	**0.21**	0.184	**0.179**
BiLSTM + CRF(**Baseline III**)	0.53	0.56	0.54	0.67	0.67	0.67	0.93	0.86	**0.89**	0.24	0.18	0.165	0.172

Table 1: Results for baselines on the ECAC dataset

pairs comes to the order of N^2, and thus we perform a random sampling of the negative pairs to keep the positive and negative samples in the ratio 1:5. This helps us to maintain balance between the positive and negative classes.

Evaluation Metrics

We evaluate the 3 steps separately as well apart from evaluating the performance for the final Emotion-Cause pairs:

Emotion Identification: We use Weighted Precision, Recall and F1-score for the distribution between the 7 classes (6 emotions and neutral class).

Candidate Cause Identification: We again use Weighted Precision, Recall and F1-score for evaluating the prediction between the binary classes – *is_candidate_cause* and *not_candidate_cause*.

Emotion-Cause Pairing: For evaluating this, we generate positive and negative pairs and use Weighted Precision, Recall and F1-score for evaluating the classification between the positive and negative classes.

Emotion-Cause Pairs: Weighted F1-score and Macro F1-score are the official metric used for the final evaluation for the task.

5 Results and Analysis

The performance of the three Baselines can be seen in Table 1. During the Evaluation phase, our best ranked submission of Baseline II had Wt. F1 score of 0.1836 and Macro F1 score of 0.1759, ranking 8th on the leaderboard.

Baseline I

Among the encoders in Baseline I, *BERT + WavLM + MViTv2* configuration performs the best on the validation set, including the individual steps as well as the final emotion-cause pair predictions. However, on the leaderboard, *EmotionRoBERTa + WavLM + MViTv2* gives the best performance, although the difference in the leaderboard scores is marginal among the encoders. This observation might indicate that the test data is a bit different in nature from the training data.

Better performance of EmotionRoBERTa can be attributed to the fact that the model's weights have already been finetuned towards emotion-related tasks. Further, it seems that finetuning DeBERTa on the training data caused it to overfit, leading to worse performance than vanilla BERT/RoBERTa models. RoBERTa-L performed slightly worse than BERT and EmotionRoBERTa.

Finally, WavLM being the newer architecture,

as expected performed better than Wav2Vec2. This is because WavLM is more robust than Wav2Vec2 and it is trained in a combination of supervised and self-supervised learning, making its performance much better.

Baseline II

We use the *EmotionRoBERTa + WavLM + MViTv2* configuration as encoders for the Baseline II architecture. Contrary to expectation, the wt. F1 score on the leaderboard decreased marginally, while the Macro F1 score increased marginally. This is probably because of the nature of the dataset, where the average length of a conversation is as little as 10, which causes the context of the utterance to be rather limited. In such a situation, the additional context from previous utterances doesn't prove helpful to the model, and might even prove to be noise for the model, leading to the results observed.

Baseline III

In this, we can observe a significant fall in wt. F1 and slight fall in Macro F1 score from the Baseline I and II architectures. This is in line with the observation of Baseline II that due to the nature of the dataset, sequence labeling it is not necessarily the best way to model it. Further, due to small number of utterances in the conversations, it is likely that the transition between labels needed for CRF doesn't get trained that well and leads to poorer performance.

6 Conclusion

In conclusion, we observe that the utterance labeling systems perform as good as sequence labeling systems for this specific dataset. Further, we also see that encoders which are trained on other emotion-related tasks tend to perform better on similar emotion-related tasks.

In future, it is possible to learn joint embeddings over the 3 modalities, which should provide better representations for each utterance (Girdhar et al., 2023). Further, it can be experimented to utilize the speaker information for each utterance while creating utterance representations (Liang et al., 2023).

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. EmoNet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 718–728, Vancouver, Canada. Association for Computational Linguistics.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Ivo Benke, Michael Thomas Knierim, and Alexander Maedche. 2020. Chatbot-based emotion management for distributed teams: A participatory design study. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–30.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Ebrahim (Abe) Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. 2008. Iemocap: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42:335–359.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 179–187, Beijing, China. Coling 2010 Organizing Committee.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3161–3170, Online. Association for Computational Linguistics.

Simon DAlfonso. 2020. Ai in mental health. *Current Opinion in Psychology*, 36:112–117.

Paul Ekman et al. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190.

Lin Gui, Ruifeng Xu, Qin Lu, Dongyin Wu, and Yu Zhou. 2016b. Emotion cause extraction, a challenging task with corpus construction. In *Social Media Processing*, pages 98–109, Singapore. Springer Singapore.

Umang Gupta, Ankush Chatterjee, Radhakrishnan Srikanth, and Puneet Agrawal. 2017. A sentiment-and-semantics-based approach for emotion detection in textual conversations. *arXiv preprint arXiv:1707.06996*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The kinetics human action video dataset.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 45–53, Los Angeles, CA. Association for Computational Linguistics.

Jingsheng Lei, Yanghui Rao, Qing Li, Xiaojun Quan, and Liu Wenyin. 2014. Towards building a social emotion detection system for online news. *Future Generation Computer Systems*, 37:438–448. Special Section: Innovative Methods and Algorithms for Advanced Data-Intensive Computing Special Section: Semantics, Intelligent processing and services for big data Special Section: Advances in Data-Intensive Modelling and Simulation Special Section: Hybrid Intelligence for Growing Internet and its Applications.

Wei Li, Yang Li, Vlad Pandelea, Mengshi Ge, Luyao Zhu, and Erik Cambria. 2022a. Ecpec: Emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1754–1765.

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. 2022b. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814.

Xingwei Liang, You Zou, and Ruifeng Xu. 2023. Silstm: Speaker hybrid long-short term memory and cross modal attention for emotion recognition in conversation. *arXiv preprint arXiv:2305.03506*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schroder. 2012. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, 3(1):5–17.

Endang Wahyu Pamungkas. 2019. Emotionally-aware chatbots: A survey.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Romila Ghosh, Niyati Chhaya, Alexander F. Gelbukh, and Rada Mihalcea. 2020. Recognizing emotion cause in conversations. *CoRR*, abs/2012.11820.

Haşim Sak, Andrew Senior, and Françoise Beaufays. 2014. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*.

Ramit Sawhney, Harshit Joshi, Alicia Nobles, and Rajiv Ratn Shah. 2021. Towards emotion-and time-aware classification of tweets to assist human moderation for suicide prevention. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 609–620.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Penghui Wei, Jiahao Zhao, and Wenji Mao. 2020. Effective inter-clause modeling for end-to-end emotion-cause pair extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3171–3181, Online. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019a. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Rui Xia and Zixiang Ding. 2019b. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

Jeewoo Yun and Jungkun Park. 2022. The effects of chatbot service recovery with emotion words on customer satisfaction, repurchase intention, and positive word-of-mouth. *Frontiers in psychology*, 13:922503.

DaVinci at SemEval-2024 Task 9: Few-shot prompting GPT-3.5 for Unconventional Reasoning

Suyash Vardhan Mathur* IIIT Hyderabad suyash.mathur@research.iiit.ac.in

Akshett Rai Jindal* IIIT Hyderabad akshett.jindal@research.iiit.ac.in

Manish Shrivastava IIIT Hyderabad m.shrivastava@iiit.ac.in

Abstract

While significant work has been done in the field of NLP on vertical thinking, which involves primarily logical thinking, little work has been done towards lateral thinking, which involves looking at problems from an unconventional perspective and defying existing conceptions and notions. Towards this direction, SemEval 2024 introduces the task of BRAINTEASER, which involves two types of questions – Sentence Puzzles and Word Puzzles that defy conventional common-sense reasoning and constraints. In this paper, we tackle both types of questions using few-shot prompting on GPT-3.5 and gain insights regarding the difference in the nature of the two types. Our prompting strategy placed us 26th on the leaderboard for the Sentence Puzzle and 15th on the Word Puzzle task.

1 Introduction

The human brain consists of two hemispheres - left and right. Both of them are responsible for different kinds of thinking strategies. The left hemisphere is involved in vertical thinking, and the right hemisphere is involved in lateral thinking (Waks, 1997). Vertical (linear, convergent, logical) thinking is a more sequential analytical process. In contrast, in Lateral (outside the box, divergent, creative) thinking, we look at the problem from a new point of view, ignoring the expected associations with items.

In the field of NLP, much research has been done around vertical thinking and significant progress has been made. The recent work around Large Language Models (LLMs) (Devlin et al., 2018; OpenAI, 2022) has achieved great performance in solving complex reasoning tasks (Talmor et al., 2018; Bisk et al., 2020; Sap et al., 2019). This performance is consistent in both cases when no task examples have been provided to the model during inference (zero-shot) (Sanh et al., 2022) and when the model is introduced with the task during inference time (few-shot) (Chung et al., 2022).

However, lateral thinking has been overlooked when training NLP models like LLMs. When creating datasets for various models, texts that involve lateral thinking are mostly considered noise and filtered out from the data because researchers want their models to perform better at traditional reasoning tasks and not get confused by lateral thinking.

The task BRAINTEASER (Jiang et al., 2023, Jiang et al., 2024) tries to bridge this gap that exists between vertical and lateral thinking for LLMs and other NLP models. They formulated a set of Multi-choice Question Answers containing puzzles that can be solved only using lateral thinking. The benchmark dataset contains two types of lateral thinking puzzles - Sentence Puzzles and Word Puzzles. This has been constructed by designing a data collection procedure that crawled relevant puzzles from many websites that were publicly available performing semi-automatic filtering of irrelevant questions.

2 Background

2.1 Dataset

The dataset being used in this task is BRAINTEASER (Jiang et al., 2023). It was prepared by scraping puzzles from various publicly available websites and then semi-automatically filtering them out. Then *semantic reconstruction* and *context reconstruction* techniques were used to create variants of each puzzle without affecting its out-of-the-box thinking style. This helped in preventing possible memorization by LLMs and the lack of consistency of the puzzles.

The puzzles in this dataset can be divided into two categories:

- **Sentence Puzzles**: These are brain teasers where the puzzle-defying commonsense is

*Equal Contribution

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1212–1216

centered on sentence snippets.

For example, **Question**: *You are running so fast but you're not getting closer. Where are you?* **Answer**: *Treadmill.* **Explanation**: This is because while running on a treadmill, we stay put where we are. The key is understanding that running on a treadmill means you remain stationary despite the motion.

- **Word Puzzles**: These are brain teasers where the answer violates the default meaning of the word and focuses more on the letter composition.

  For example, **Question**: *How can you make "ten" out of "net"?* **Answer**: *Just flip it around.* **Explanation**: This is because if we consider the spelling of the word "ten" and we flip the letters of the word around, we get the word "net" which is what we want to make out of "ten".

The training data contains **507 Sentence Puzzles** and **396 Word Puzzles**. Each of these puzzles has 4 options to choose from and only one option is the correct answer.

2.2 Related Works

With the recent success of LLMs in various NLP tasks, researchers have also started exploring their use for Multiple Choice Question Answering (MCQA) tasks (Robinson et al., 2022; Zheng et al., 2023).

Researchers have also started employing the technique of **few-shot prompting** (Liu et al., 2023; Ma et al., 2024; Lu et al., 2021) for various tasks and it has shown improvements when compared with **zero-shot prompting**.

LLMs like GPT-3.5 have been trained on vast amounts of human-generated text. The main features around which such models are trained are **Pattern Recognition**, **Creative Reasoning** and **Wide Knowledge Range**.

Thus, we decided to employ few-shot prompting on LLMs for this task.

3 System Overview

Our architecture uses GPT-3.5 (Brown et al., 2020) (specifically *gpt-3.5-turbo*) with few-shot prompting to answer the question.

3.1 GPT-3.5

In NLP, the architecture of Generative Pre-trained Transformer (GPT) 3.5 (GPT-3.5) stands as a significant advancement, which is the culmination of iterative improvement over its predecessors. The architecture of the model is based upon the Transformer model (Vaswani et al., 2017), which uses self-attention to enhance performance over the prior sequential models. GPT-3.5 scales this Transformer architecture to over hundereds of billions of parameters, which have been trained by exposing and training the model on hundreds of billions of tokens.

In particular, due to the autoregressive nature of GPT-3.5 and due to being trained on extremely large data, it has enough knowledge about the language and the real world to perform tasks in a Zero-shot setting (Sanh et al., 2022). This Zero-shot setting allows the model to understand and execute a task it hasn't been explicitly trained for. These capabilities have been reflected in GPT-3.5 being used in Summarization (Liu and Healey, 2023), Question Answering (Bahak et al., 2023), Natural Language Inference (Ye et al., 2023), etc.

3.2 Few-shot prompting

While zero-shot prompting works well for simple tasks, tasks like BrainTeaser are a bit more complex in nature, and in such cases providing explicit instructions to the LLM about the nature of the task along with few examples of the task *(few shots)* becomes extremely helpful for the model (Chung et al., 2022). Here, the few-shot technique involves providing GPT-3.5 some examples, allowing GPT-3.5 to generalize from the few examples, drawing on its large pre-trained knowledge about the language and the real-world.

Thus, 2 different sets of prompts are created for the task, one for the Sentence Puzzle Task and another for the Word Puzzle task, since the 2 tasks are fundamentally different and need different instructions and examples.

3.3 Experimental Setup

We provide 2-shot prompts to GPT-3.5 for our leaderboard submission. We also try out 5-shot prompt in the post evaluation phase to test if providing more examples helps the model perform better.

The prompt used for the Sentence Puzzle task is shown in Listing 1. As we can see, the prompt first

```
You are given a question with multiple choices that you need to answer.
↪  The answer would only be one index of the multiple choices available.
↪  Such a question would involve  brain teaser questions where the
↪  puzzle defying commonsense is centered on sentence snippets.
IMPORTANT: It's crucial to analyze the question from an unconventional
↪  perspective, focusing on the literal or alternative meanings of the
↪  words used, rather than relying on common sense. You must not use
↪  commonsense, but look at meaning from a different perspective than
↪  what would commonly be done. For example,

Example 1:
Question: You are running so fast but you're not getting closer. Where
↪  are you?

Option 0: Country road.
Option 1: Treadmill.
Option 2: High way.
Option 3: None of above.

Answer: 1
Reason: This is because while running on a treadmill, we stay put where
↪  we are. The key is understanding that running on a treadmill means
↪  you remain stationary despite the motion. This is not valid for
↪  Country road or High way. Thus, the answer is 1 - Treadmill.

Example 2:
Question: From elementary school to collage, how many "first day of
↪  school" does the average person have in their lifetime?

Option 0: They technically only have one first day of school in their
↪  lifetime. That's the very first day they started attending school
↪  as a child.
Option 1: Average people have 4: elementary school, middle school, high
↪  school, and college.
Option 2: Average people have "first day of school" in each semester,
↪  so it will be more than 10!
Option 3: None of above.

Answer: 0
Reason: First day of school can only be one day in a person's lifetime.
↪  Here, it is important to understand that first day of middle school,
↪  high school, college won't be first day of school. Similarly, each
↪  semester's first day is not TECHNICALLY first day of school. This,
↪  the answer is 0 -  They technically only have one first day of
↪  school in their lifetime. That's the very first day they started
↪  attending school as a child. Thus, the key here is the term 'first
↪  day of school' technically refers to the very first day a person
↪  attends school, making all subsequent 'first days' at different
↪  educational levels irrelevant to the specific question.

Now, using these examples, answer the question below. It is IMPORTANT
↪  that you just provide the index of the answer in the response. DO
↪  NOT output the reason behind choosing the answer:

Question: In a small village, two farmers are working in their fields -
↪  a diligent farmer and a lazy farmer. The hardworking farmer is the
↪  son of the lazy farmer, but the lazy farmer is not the father of
↪  the hardworking farmer. Can you explain this unusual relationship?
Option 0: The lazy farmer is his mother.
Option 1: The lazy farmer is not a responsible father as he is lazy.
Option 2: The diligent farmer devoted himself to the farm and gradually
↪  forgot his father.
Option 3: None of above.

Answer:
```

Listing 1: Prompt for the Sentence Puzzle

details the task, and *IMPORTANT* keyword is used to express to GPT-3.5 that commonsense must not be used in the task, but instead it should look at meaning from an unconventional sense. Then, 2 examples are given, along with **reasoning** behind the answers too. This was important, as this gave the model more knowledge to be able to generalize the task from the examples. Further, the output format was clearly specified in the prompt so as to avoid getting extra information in the model output.

Similar prompt for Word Puzzle can be seen in Listing 2. The prompt clarifies that the structural aspect of the words should be focused on, emphasizing that unconventional meaning should be looked at. Then, 2 examples that exhibit structural aspect are given along with reasoning behind their answers as well as constraints for the output format.

```
You are given a question with multiple choices that you need to answer.
↪  The answer would only be one index of the multiple choices
↪  available. The question demands an unorthodox approach, focusing on
↪  the spellings or structural aspects of words, rather than their
↪  standard meanings. Your task is to choose the correct answer from
↪  the given multiple-choice options by analyzing the words in a
↪  literal or unconventional way.
IMPORTANT: It's crucial to analyze the question from an unconventional
↪  perspective, focusing on the spellings of certain words, rather
↪  than relying on common sense. You must not use commonsense, but
↪  look at meaning from a different perspective considering
↪  arrangement of the letters in certain words than what would
↪  commonly be done. For example,

Example 1:
Question: How can you make "ten" out of "net"?

Option 0: Just flip it around.
Option 1: Remove the letter "e".
Option 2: Move the letter "t" to the end.
Option 3: None of above.

Answer: 0
Reason: This is because if we consider the spelling of the word 'ten'
↪  and we flip the letters of the word 'ten' around, we get the word
↪  'net', which is what we want to make out of 'ten'.  The answer
↪  focuses on the literal rearrangement of the letters, disregarding
↪  the typical meanings of the words. Thus, the answer is 0 - Just
↪  flip it around.

Example 2:
Question: What is the most fast city?

Option 0: Urban city.
Option 1: Inner city.
Option 2: Velocity.
Option 3: None of above.

Answer: 2
Reason: The term 'fast' in the question prompts an unconventional
↪  interpretation. All options contain the word "city", but "velocity"
↪  stands out as it directly relates to speed or 'fastness'. The
↪  question cleverly uses the term 'city' as a red herring, while the
↪  actual focus is on the concept of speed.

Now, using these examples, answer the question below. It is IMPORTANT
↪  that you just provide the index of the answer in the response. DO
↪  NOT output the reason behind choosing the answer:

Question: What sort of cheese is made in reverse?
Option 0: Cheddar cheese..
Option 1: Edam cheese.
Option 2: Blue cheese.
Option 3: None of above.

Answer:
```

Listing 2: Prompt for the Word Puzzle

4 Results and Analysis

The results are detailed in Tab.1. For comparison, we also list the zero-shot prompting results reported in Jiang et al., 2023. As we can see, the two-shot performance on Word puzzle improved over the zero-shot setting for all the categories, while the same worsened in case of the Sentence puzzle.

This is because of the very nature of the two problems. Sentence puzzle involves deeper non-

Table 1: Results of zero-shot and few-shot prompting on GPT-3.5 for the two BRAINTEASER subtasks. Ori = Original, Sem = Semantic, Con = Context.

Model	Instance-based			Group-based		Overall
	Original	Semantic	Context	Ori & Sem	Ori & Sem & Con	
Sentence puzzle						
GPT-3.5 (zero-shot) **Baseline**	60.7	59.3	**67.9**	50.7	39.7	**62.6**
GPT-3.5 (two-shot)	57.5	55.0	42.5	50.0	30.0	51.7
GPT-3.5 (five-shot)	**62.5**	**65.0**	55.0	**62.5**	**42.5**	60.8
Word puzzle						
GPT-3.5 (zero-shot) **Baseline**	56.1	52.4	51.8	43.90	29.3	53.5
GPT-3.5 (two-shot)	71.9	71.9	62.5	59.4	46.9	68.6
GPT-3.5 (five-shot)	**78.1**	**90.6**	**84.4**	**78.1**	**68.8**	**84.4**

conventional semantic understanding of the question and the choices, which despite conveying reasoning behind the answers in the few-shot examples, cannot be generalized as easily with just 2 examples. On the other hand, the only tricky component of the Word Puzzle is that the structural aspect of certain words needs to be taken instead of the actual surface meaning of the said words. This can be much more easily generalized through just as few as two examples in the prompt. Further, adding the examples in the Sentence puzzle that don't generalize very well for other questions in the testing set might have acted as noise for the model, which led to poorer performance.

We also note that using five-shot prompt instead of two-shot prompt hugely increases the performance. This is to be expected, as providing more examples would help the model generalize even better towards solving the task. This is specially true for Word Puzzle questions, where adding more examples allows the model to generalize the task much better.

However, in Sentence Puzzle we still notice a drop in the overall performance as compared to the zero-shot model. This is because of a drop in the performance of the context reconstruction questions, and a marginal increase in comparison to zero-shot in other types of questions. However, group based accuracy increases in five-shot, which might indicate that with five examples, the model is able to handle the variations in reconstructions better, albeit with performance of Contextual Reconstruction taking a dip. These observations are in line with the drop observed in two-shot prompt in comparison to the zero-shot prompt, highlighting the difficult nature of the task of Sentence Puzzle questions and the inability of the model to generalize using few Sentence Puzzle examples. However, we do note that the performance on Sentence Puzzle also does improve with additional examples between two-shot and five-shot prompting.

5 Conclusion

In conclusion, we explored the effectiveness of few-shot prompting for LLMs for complex and unconventional tasks. Further, it demonstrates that few-shot prompting is helpful only in scenarios where the examples convey enough information that can be better generalized, as the results worsened in the Sentence Puzzle while improved in the Word Puzzle.

In future, better prompting strategies like Chain of Thought prompting (Wang et al., 2023) can be utilized to improve the performance. Additionally, finetuning the pre-trained LLMs might also help in the task further. Also, increasing the number of training examples might help in further improving the model performance, as observed in the gains of performance in the five-shot prompt in comparison to the two-shot prompt.

References

Hossein Bahak, Farzaneh Taheri, Zahra Zojaji, and Arefeh Kazemi. 2023. Evaluating chatgpt as a question answering system: A comprehensive analysis and comparison with existing models.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish,

Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Sengjie Liu and Christopher G. Healey. 2023. Abstractive summarization of large document collections using gpt.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Huan Ma, Changqing Zhang, Yatao Bian, Lemao Liu, Zhirui Zhang, Peilin Zhao, Shu Zhang, Huazhu Fu, Qinghua Hu, and Bingzhe Wu. 2024. Fairness-guided few-shot prompting for large language models. *Advances in Neural Information Processing Systems*, 36.

OpenAI. 2022. Chatgpt: A language model by openai. `https://www.openai.com`.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2022. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6:245–255.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. On large language models' selection bias in multi-choice questions. *arXiv preprint arXiv:2309.03882*.

MorphingMinds at SemEval-2024 Task 10: Emotion Recognition in Conversation in Hindi-English Code-Mixed Conversations

MONIKA VYAS
Purdue University Fort Wayne
vyasm01@pfw.edu

Abstract

The complexity of expressing emotions in multilingual settings, particularly in Hindi-English code-mixed conversations (Bafna and Gali, 2022), presents both obstacles and prospects for natural language processing (NLP) research. This thesis ventures into the realm of emotion recognition within code-mixed text (Sasidhar et al., 2022), to enhance comprehension and technological capabilities in this domain.

The principal objective of this study is to refine NLP models tailored specifically to the intricacies of code-mixed Hindi-English conversations. By harnessing advanced deep learning architectures (Sane et al., 2019a) like BERT, RoBERTa, and BERTweet, the research systematically evaluates the efficacy of various models in capturing nuanced emotional expressions embedded within code-mixed text.

Utilizing data from the EDiReF shared task at SemEval 2024 (Kumar et al., 2024a), the dataset encompasses dialogues sourced from a popular Indian comedy television series, offering a diverse range of conversational excerpts reflecting cultural nuances and comedic elements. Through meticulous data analysis and preprocessing, insights into the distribution of emotions and linguistic patterns within the dataset are gleaned, informing subsequent model selection and training strategies.

The process of model selection adopts an iterative approach, commencing with traditional machine learning models such as Support Vector Machines (SVM) and Logistic Regression (LR) before transitioning to deep learning architectures like XLM-BERT. Techniques for model training and optimization evolve, integrating validation datasets to assess generalization capabilities and ensuring robust evaluation methodologies.

Feature extraction methods, including TF-IDF vectorization (Mikolov et al., 2023) and N-gram analysis, are employed to capture pertinent linguistic patterns and contextual information from the text data, thereby enriching the representation of textual features (Feng and Liu, 2021) crucial for emotion detection.

In summary, this thesis contributes to the advancement of emotion recognition technology in code-mixed languages (Gupta et al., 2022), shedding light on the intricate interplay of emotions within Hindi-English conversations. By addressing the unique challenges posed by code-mixed languages and harnessing state-of-the-art NLP techniques, this research lays the groundwork for applications in sentiment analysis, conversational AI, and cross-cultural communication.

1 Introduction

In this thesis, we explore the topic of Emotion Recognition in Conversation in Hindi-English Code-Mixed Conversations. The research is motivated by the challenges addressed in the SemEval 2024 Task 10: Emotion Discovery and Reasoning its Flip (Kumar et al., 2022) (Kumar et al., 2024b) in Conversation (EDiReF) (Kumar et al., 2024a).

This paper addressed the Emotion Recognition in Conversation (ERC) task one (Kumar et al., 2023), one of the three subtasks in the Emotion Discovery and Reasoning its Flip in Conversation (Kumar et al., 2022) (Kumar et al., 2024b). ERC involves assigning emotions to each utterance in a dialogue from a predefined set of possible emotions. The research specifically focuses on ERC to contribute to advancing emotion recognition and understanding in multilingual and code-mixed conversational settings.

The initial stages of the research endeavor encompassed several pivotal tasks aimed at laying the foundation for exploring emotion detection within multilingual conversational contexts. Foremost among these tasks was the meticulous preprocessing of the dataset, which involved addressing anomalies such as missing speaker information (NaN) in the JSON files. Additionally, developed a

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1217–1221

Speaker	Utterance	Emotion
Sp_1	Aaj to bhot awful day tha! *(I had an awful day today!)*	Sad
Sp_2	Oh no! Kya hua? *(Oh no! What happened?)*	Sad
Sp_1	Kisi ne mera sandwich kha liya! *(Somebody ate my sandwich!)*	Sad
Sp_2	Me abhi tumhare liye new bana deti hun! *(I can make you a new one right now!)*	Joy
Sp_1	Wo great hoga! Thanks! *(That would be great! Thanks!)*	Joy

Figure 1: ERC dataset format (Kumar et al., 2023)

common module for flattening lists and organized preprocessing code into separate modules to enhance efficiency and facilitate code reusability.

A critical aspect of the methodology involved harnessing transformer models pre-trained in Romanized Hindi for fine-tuning efforts. Particularly noteworthy was the fine-tuning of a pre-trained XLM-RoBERTa model using a custom dataset. This adaptation was necessitated by the absence of labels in a compatible format with the XLM model's training corpus. The fine-tuning process yielded promising outcomes, evidenced by the training output demonstrating a final training loss of approximately 0.3471 and robust training efficiency metrics such as runtime and samples per second.

However, challenges emerged during the subsequent phases of model evaluation and prediction. An initial attempt to predict emotions using the development dataset and the fine-tuned model yielded unexpected results, with all predictions aligning with the "neutral" label. Subsequent analysis revealed imbalances in label distribution, prompting a meticulous review of label mapping and encoding procedures to ensure the accuracy of model predictions.

In response to these challenges, alternative strategies were explored, including the augmentation of the number of epochs in model training. Despite these efforts, evaluation metrics following this adjustment showed marginal improvements, with precision, recall, and F1-scores remaining low across various emotion categories.

Further experimentation involved ensemble techniques such as bagging and boosting algorithms, implemented through classifiers like Random Forest and AdaBoost. While these approaches demonstrated some enhancement in performance metrics, challenges persisted, particularly in classes with low precision and recall. A pivotal juncture in the research journey involved the exploration of oversampling techniques, inspired by insights from the literature highlighting the limitations of undersampling in small datasets. Techniques such as Synthetic Minority Over-sampling Technique (SMOTE) [16] were considered to address the class imbalance and enhance model robustness. Additionally, attention was given to hyperparameter adjustments, including batch size, learning rate, and model architecture, to optimize model performance.

2 Related Work

Following is research on similar Emotion detection in conversations with multiple labels and multilingual language : Emotion detection and classification have been extensively explored within monolingual datasets. However, the challenge of dealing with code-mixed text, particularly in languages like Hindi combined with English, has resulted in fewer studies in this domain. Notably, Vijay et al. (Vijay et al., 2018) conducted seminal research on emotion detection in social media text characterized by Hindi-English code-mixing (Chauhan et al., 2019). They curated a corpus comprising 2866 sentences across six emotion classes and conducted experiments focusing on three classes: Happy, Sad, and Anger. Their methodology involved preprocessing the data, extracting character-n-grams and word-n-grams as primary features, employing chi-square for feature selection, and employing a Support Vector Machine (SVM) as the classifier, achieving an accuracy of 58 percent.

In a similar vein, Ghosh et al. (Ghosh et al., 2017) undertook sentiment detection tasks on code-mixed text extracted from social media platforms, utilizing English-Bengali and English-Hindi datasets. Their approach involved classifying sentences based on polarity contradictions, leveraging features such as Sentiwordnet word matches, opinion lexicons, and POS tags. They employed a Multilayer Perception model, achieving an accuracy of 68.5 percent.

Joshi et al. (Prabhu et al., 2016) explored sentiment analysis in Hindi-English code-mixed text sourced from Facebook comments, maintaining a polarity scale and forming a corpus of 3879 sentences across three classes. Their unique classification method involved sub-word level LSTM (Hochreiter and Schmidhuber, 1997), outperforming traditional algorithms like Char-LSTM, SVM-Unigram, and Naive Bayes, with an accuracy of 69.7 percent.

A survey conducted by Samar et al. (Al-Saqqa et al., 2018) categorized four different approaches to emotion classification: keyword-based, corpus-based, learning-based, and hybrid approaches. The the survey highlighted the efficacy of hybrid approaches, particularly ensemble techniques, and emphasized the promising outcomes of deep learning models.

Additionally, Shalini et al. (Shalini et al., 2019) explored stance detection in English-Kannada code-mixed data, utilizing deep learning architectures (Tripathi et al., 2013) and text representations such as Word2Vec and GloVe. Their findings showcased the effectiveness of CNN in learning new weights on top of a pre-trained model.

Further, studies by Sane et al. (Sane et al., 2019b), and Satyajit et al. (Kamble and Joshi, 2018) delved into aggression detection, humor detection, and hate speech detection in code-mixed data, respectively, employing various techniques such as text-based features, fastext embeddings, and bilingual embeddings generated using Word2Vec.

Reflecting on the reviewed literature, it becomes apparent that various machine learning models, such as Support Vector Machines (SVM), Logistic Regression, Naive Bayes, and Transformer-based models like XLM-RoBERTa (Wei, 2021), offer valuable features for model learning in emotion classification tasks. While deep neural networks, particularly those incorporating CNN as a primary layer, have exhibited superior performance in some studies, the approach focused on leveraging a diverse set of machine learning algorithms. In alignment with these findings, the study adopts different machine learning models and transformer architectures, including SVM, Logistic Regression, and XLM-Roberta, for emotion classification in code-mixed text.

3 Methodology

We initiated the data pre-processing phase to ensure the quality and relevance of the dataset. Initially, the dataset contained 3475 comments alongside unrelated information, prompting the creation of a refined dataset focusing solely on relevant information such as utterances and their corresponding emotions. In the pursuit of model development, we commenced by employing traditional machine learning (ML) models, including Logistic Regression (LR), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM). Despite initial efforts, the obtained accuracy of 10 percent and F1-score were unsatisfactory 9 percent, with all predicted labels converging to neutral across the test utterances.

Subsequently, we transitioned to fine-tuning a pre-trained XLM-Roberta model tailored to handle the Romanized Hindi language. We leveraged a custom dataset due to its unique label format incompatible with the XLM model's training corpus. The training process yielded promising results, with a final training loss of approximately 0.3471. Key training metrics, including runtime, samples processed per second, and total FLOPs, underscored the efficiency and effectiveness of the training procedure.

Furthermore, we explored alternative approaches to address the neutral label prediction issue, drawing insights from external resources such as a referenced medium article and relevant research papers. Experimentation with TF-IDF feature extraction revealed discrepancies in data formatting, necessitating adjustments to ensure accurate feature extraction. By rectifying these data pre-processing issues and revisiting logistic regression, we successfully diversified predicted labels across utterances, albeit with reduced accuracy and F1-score.

To address the dataset's imbalance and size constraints, we delved into oversampling techniques, particularly the Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN) (Maharana and Mohapatra, 2021). By oversampling the dataset and optimizing TF-IDF parameters, we observed improvements in model performance, achieving an accuracy of 33 percent and an F1 score of 35 percent.

In the adjusted formula in Equation 1 below: TP, TN, FP, and FN symbolize the counts of true positives, true negatives, false positives, and false negatives, respectively. These metrics serve as pivotal indicators in assessing the model's efficacy.

$$\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} + \sum_{i=1}^{n} \frac{\text{Synthetic Samples}_i}{\text{Original Dataset}}$$

Figure 2: Model Performance

Precision and Recall encapsulate the model's acumen in delineating emotional instances within the dataset accurately. They provide insights into the model's capability to discern and categorize

emotions effectively. The summation term of 1nOriginal Dataset- Synthetic SamplesSMOTE elucidates the influence of SMOTE oversampling on the dataset.

In this context, the Original Dataset denotes the size of the initial dataset, while Synthetic SamplesSMOTE signifies the number of synthetic samples generated through SMOTE oversampling. This component reflects SMOTE's remedial impact on rectifying class imbalances within the dataset, thereby fostering enhanced model performance.

Model			Accuracy
SVM			0.30
Logisitc Regression			0.33
XLM-Roberta			0.20
Precision	**Recall**	**F1 Score**	
0.32	0.30	0.30	
0.38	0.33	0.35	
0.25	0.20	0.28	

Table 1: Model Performance Metrices

4 Results and Discussion

The results presented in Table 1 provide valuable insights into the performance of different models in emotion recognition. Notably, the evaluation metrics—accuracy, precision, recall, and F1-Score—offer a comprehensive view of each model's effectiveness in capturing the nuances of emotional expressions within the dataset.

One striking observation is the superior performance of logistic regression (LR) compared to Support Vector Machine (SVM) and transformer-based models like XLM-Roberta. LR outperformed SVM and XLM-Roberta.

with an accuracy of 33 percent, showcasing its ability to better generalize to the dataset and make accurate predictions. This outcome is particularly noteworthy given the challenges posed by small datasets, where traditional machine learning models often excel due to their simplicity and interpretability. The utilization of SMOTE oversampling and TF-IDF feature engineering played a pivotal role in enhancing model performance. By augmenting the dataset through SMOTE and increasing the maximum number of features in TF-IDF from 5000 to 8000, effectively addressed class imbalance and enriched the feature space, leading to a notable improvement in accuracy and F1-Score. This underscores the importance of preprocessing techniques in mitigating dataset constraints and improving model robustness.

Furthermore, the decision to prioritize logistic regression over transformer-based models reflects the nuanced requirements of emotion recognition tasks in small datasets. While transformer-based models are celebrated for their ability to capture complex patterns in large datasets, their performance may be suboptimal in scenarios where data scarcity is prevalent. By leveraging logistic regression, striked a balance between model complexity and dataset size, resulting in more reliable and interpretable emotion recognition systems. Overall, these findings reaffirm the efficacy of traditional machine learning approaches in handling emotion recognition tasks, especially when confronted with limited data availability. Moving forward, exploring hybrid models that integrate the strengths of both traditional machine learning and transformer-based architectures could pave the way for even greater advancements in emotion understanding and recognition.

5 Conclusion and Future Work

In the forthcoming research endeavors, we aim to expand the scope of the emotion detection framework by integrating both word embeddings and TF-IDF features. This innovative approach seeks to create a richer representation of textual data by combining semantic embeddings with feature importance weighting.

The primary focus will be on harnessing Convolutional Neural Network (CNN) architectures to further refine the emotion detection process. Through the training and optimization of the CNN model using this blended feature space, we anticipate significant enhancements in the model's capacity to discern subtle emotional nuances within the text. To validate the effectiveness of this approach, we will employ standard performance metrics and conduct comparative analyses against baseline models.

Additionally, we envision extending the methodology to accommodate multimodal data sources, such as text paired with audio or visual inputs (Dhawan and Wadhawan, 2022). This expansion will serve to broaden the application of emotion detection, opening avenues for more comprehensive analyses and interpretations of emotional content across various media formats.

In summary, the research journey has been characterized by iterative experimentation and adap-

tation in response to emerging challenges and insights. The endeavors underscore the complexity inherent in emotion detection within multilingual conversational data and emphasize the significance of methodological rigor and innovation in overcoming these challenges. Moving forward, the focus remains on refining methodologies and exploring novel approaches to further enhance the accuracy and robustness of emotion detection systems in diverse linguistic and cultural contexts.

References

Samar Al-Saqqa, Heba Abdel-Nabi, and Arafat Awajan. 2018. A survey of textual emotion detection. In *8th International Conference on Computer Science and Information Technology (CSIT)*, pages 136–142. IEEE.

Abhishek Bafna and Karthik Gali. 2022. Dravidian language technology: Some perspectives.

Nidhi Chauhan, Shubham Atreja, Anubhav Garg, and Prakhar Gupta. 2019. Emotion detection in hinglish/hindi/english code-mixed social media text.

Priya Dhawan and Arnav Wadhawan. 2022. Multi-head attention: What it is and how to use it.

Zhe Feng and Bing Liu. 2021. A survey of textual emotion detection.

Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*.

Sanket Gupta, Monika Sharma, and Rajeev Jain. 2022. Sentiment identification in code-mixed social media text.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory.

Satyajit Kamble and Aditya Joshi. 2018. Hate speech detection from code-mixed hindi-english tweets using deep learning models. *arXiv preprint arXiv:1811.05145*.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024a. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2024b. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, 5(3):1339–1348.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Trideep Maharana and Himansu Mohapatra. 2021. Impact of smote on imbalanced text features for toxic comments classification using rvvc model.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2023. Efficient estimation of word representations in vector space.

Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. *arXiv preprint arXiv:1611.00472*.

Sushmitha Reddy Sane, Suraj Tripathi, Koushik Reddy Sane, and Radhika Mamidi. 2019a. Deep learning techniques for humor detection in hindi-english code-mixed tweets.

Sushmitha Reddy Sane, Suraj Tripathi, Koushik Reddy Sane, and Radhika Mamidi. 2019b. Deep learning techniques for humor detection in hindi-english code-mixed tweets. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 57–61.

Turaga Tulasi Sasidhar, Premjith B, and Soman Kp. 2022. Emotion detection in hinglish(hindi+english) code-mixed social media text.

K Shalini, M Anand Kumar, and K Soman. 2019. Deep-learning-based stance detection for indian social media text. In *Emerging Research in Electronics, Computer Science and Technology*, pages 57–67. Springer.

Suraj Tripathi, Aditya Joshi, and Radhika Mamidi. 2013. Aggression detection on social media text using deep neural networks.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Corpus creation and emotion prediction for hindi-english code-mixed social media text. In *Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: student research workshop*, pages 128–135.

Chen Wei. 2021. Train an xlm-roberta model for text classification on pytorch.

SemanticCUETSync at SemEval-2024 Task 1: Finetuning Sentence Transformer to Find Semantic Textual Relatedness

Md. Sajjad Hossain, Ashraful Islam Paran, Symom Hossain Shohan, Jawad Hossain and **Mohammed Moshiul Hoque**
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
{u1904031,u1904029,u1904048,u1704039}@student.cuet.ac.bd
moshiul_240@cuet.ac.bd

Abstract

Semantic textual relatedness is crucial to Natural Language Processing (NLP). Methodologies often exhibit superior performance in high-resource languages such as English compared to low-resource ones like Marathi, Telugu, and Spanish. This study leverages various machine learning (ML) approaches, including Support Vector Regression (SVR) and Random Forest, deep learning (DL) techniques such as Siamese Neural Networks, and transformer-based models such as MiniLM-L6-v2, Marathi-sbert, Telugu-sentence-bert-nli, and Roberta-bne-sentiment-analysis-es, to assess semantic relatedness across English, Marathi, Telugu, and Spanish. The developed transformer-based methods notably outperformed other models in determining semantic textual relatedness across these languages, achieving a Spearman correlation coefficient of 0.822 (for English), 0.870 (for Marathi), 0.820 (for Telugu), and 0.677 (for Spanish). These results led to our work attaining rankings of 22th (for English), 11th (for Marathi), 11th (for Telegu) and 14th (for Spanish), respectively.

1 Introduction

Semantic textual relatedness measures the conceptual and contextual similarity of two sentences. It specifies how alike the two sentences are in terms of meaning. Determining semantic textual relatedness is crucial for various language-processing tasks, including contemporary technology, search engines, chatbots, virtual assistants, plagiarism detection, paraphrasing, question answering, text generation, and other related applications. It is possible to determine how comparable two natural language sentences are based on the quantity and quality of matched elements in each sentence. These matches offer essential insights into the relationship between and degree of semantic similarity between the two sentences and the likelihood of successful word matching in semantically equivalent text pairs.

Another critical aspect of semantic relatedness is understanding the context of sentences. Singnificant research has been done in this area, specifically in SemEval competition from 2012 to 2017 (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017). Most proposed methodologies perform better in high-resource languages like English and Spanish but could be better in other low-resource languages like Telegu, Marathi, and Bengali. In certain studies, translating from a low-resource language to English is conducted (Wu et al., 2017) prior to semantic analysis, contributing to suboptimal performance. Other factors include the failure of specific methods to capture the meaning of phrases and idioms within sentences effectively (Śpiewak et al., 2017), as well as a tendency for some approaches to focus excessively on individual word meanings (Śpiewak et al., 2017). This work addresses these shortcomings by proposing a transformer-based model, which gives us a good result in finding semantic textual relatedness. The main contributions of this work are:

- Investigated several ML, DL, and transformer-based models to find the semantic relatedness in various texts of four languages.
- Explored various pre-trained transformer-based methods with necessary tuning to identify semantic textual relatedness in English, Spanish, Telegu, and Marathi.

The code will be publicly available at https://github.com/ashrafulparan2/SemanticCUETSync-at-SemEval-2024-Task-1.

2 Related Work

Numerous studies have been accomplished on semantic relatedness in high-resource languages. Hasan et al., 2020 presented the process for calculating semantic similarity and proposed a feature-based metric for building semantic vectors. Their

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1222–1228

knowledge feature-based method found similarity measure of 0.82. Abdalla et al., 2021 proposed a dataset to explore questions on what makes sentences semantically related. Reimers and Gurevych, 2019 suggested using a sentence transformer model, which creates a 768-dimensional dense vector space from sentences and paragraphs. The relatedness of two sentences can be assessed using this model. Their approach achieved the best score of 0.8492 with SRoBERTa-STSb-base model. Three transformer-based clinical semantic textual similarity models intended to detect semantic relatedness in medical data were presented by Yang et al., 2020. Chen et al., 2022 proposed a semi-supervised sentence embedding technique called GenSE, which effectively uses large-scale unlabeled data. It achieved promising results on several STS datasets with an average correlation score of 0.8519. Meanwhile, Gatto et al., 2023 evaluated sentence similarity among texts using large language models (LLM). According to their research, ChatGPT and other models are proficient in recognizing textual similarity within particular areas. In addition to the transformer models, Verma and Muralikrishna, 2020 introduced a deep learning model, specifically a Recurrent Neural Network (RNN). This model utilized document embedding vectors to infer the meaning of small paragraphs comprising one, two, or three sentences.

Significant challenges arise when representing sentences with low-resource languages (LRLs), such as Telugu, Marathi, and Bengali. Furthermore, limited datasets make the process of textual similarity detection more challenging in LRLs. Joshi et al., 2023 focused on two low-resource Indian languages (Hindi and Marathi), and their proposed model was evaluated on real text classification datasets to show embeddings obtained from synthetic data, which will be an effective training strategy for low-resource languages. They achieved the highest score of 0.83 utilizing the MahaBERT and LaBSE models. A cross-lingual model for finding similarity between sentences was proposed by Deode et al., 2023a. Their system obtained an accuracy of 0.82 for finding semantic relatedness in low-resource languages like Hindi, Marathi, Kannada, Telugu, Malayalam, Tamil, Gujarati, Odia, Bengali, and Punjabi. Tang et al., 2018 proposed a multilingual framework for finding semantic textual similarity in low-resource languages utilizing rich annotation data from a high-resource language. Their shared sentence encoder approach archived score of 0.825 for Spanish language.

3 Dataset and Task Description

The dataset is developed by Ousidhoum et al. (Ousidhoum et al., 2024a) to evaluate the semantic textual similarity to perform the shard task at SemEval-2024. It includes data for Telugu, Marathi, English, and Spanish, which assess semantic relatedness between sentences. There are labeled and unlabeled data in the dataset. Labeled data is used for Track-A; each row has two sentences and a score that ranges from 0 to 1, representing the semantic relatedness of the sentences. Moreover, the dev, test, and train categories are applied to every language dataset. Table 1 shows the distribution of dev, test, and train sets for the English, Marathi, Telugu, and Spanish datasets, respectively.

Language	Dev set	Test set	Train set
English	250	2600	11000
Marathi	294	299	2400
Telegu	130	297	2340
Spanish	140	600	1562
Total	814	3796	17302

Table 1: Dataset statistics for Track-A

The task for Track-A (Ousidhoum et al., 2024b) calculates the semantic relationship and provides a score (degree of semantic relatedness) between 0 to 1. Figure 1 illustrates few examples of the task of sentence relatedness with the score.

4 System Overview

Various textual features will be extracted to employ ML and DL models. Several transformer models are exploited for the task, including MiniLM-L6-v2, Marathi-SBERT, Telugu-Sentence-BERT-NLI, and Roberta-BNE-Sentiment-Analysis-ES, to assess semantic relatedness. Figure 2 illustrates the schematic structure of the proposed approach.

Textual Feature Extraction: Feature extraction is necessary for ML and DL models to learn from text. We used TF-IDF (Takenobu, 1994) to extract the features to apply different ML algorithms. Word2Vec (Mikolov et al., 2013) and FastText (Grave et al., 2018) embeddings were used to extract features for DL models.

ML-based Approaches: This work employed traditional ML approaches, including SVR and RF. Following dataset preprocessing, we trained

Sentence #1	Sentence #2	Score
I very much enjoy Billy Talent. They are one of my favorite groups.	I just love Billy Talent, they are one of my favourites.	0.97
El cocinero estÃ¡ rociando queso sobre la pizza.	Un hombre estÃ¡ poniendo un poco de queso en una pizza.	0.57
పశ్చిమబంగలో రైల్వే ఇరుసు కర్మాగారాన్ని నెలకొల్పనున్నట్లు ఆయన వెల్లడించారు	ఈ దాడులు రాష్ట్రవ్యాప్తంగా కొనసాగుతాయని ఈ సందర్భంగా విజిలెన్స్ డీజీ స్పష్టం చేశారు.	0.37
या विश्वचषकानंतर जयसूर्याने मागे वळून पाहिले नाही.	त्यातूनच तिने इसिसकडे भारतातील मुस्लिमांना आकृष्ट करण्याचे काम सुरू केले.	0.04

Figure 1: Track-A task sample with Similarity score for English, Spanish, Telugu and Marathi

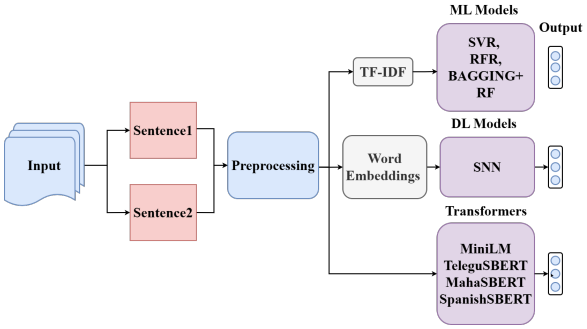


Figure 2: Schematic process of finding semantic textual relatedness

the model using the SVR for the English language. Similarly, we utilized the RF, configuring "n-estimators = 100" during training. A BaggingRegressor model is employed with the RF as the base estimator, with "n-estimators = 10" set for the base estimator.

DL-based Approaches: This work explored a Siamese Neural Network (SNN) architecture implemented in PyTorch to perform the task. The SNN model was applied exclusively to the English dataset. The text is transformed into vectors and subsequently passed through LSTM layers. The similarity between the two texts was determined using cosine embedding loss, and optimization was carried out using the 'Adam' optimizer. We set learning rate = 10, hidden-dim = 128, embedding-dim = 128 and max-length = 1000 and ran it for 10 epochs. Table 2 shows the several ML parameters and DL hyperparameters used for the models.

Classifier	Parameters	Value
SVR	kernel	rbf
	degree	3
	gamma	scale
	C	1.0
RF	n-estimator	100
Bagging+RF	n-estimator	10
SNN	epoch	10
	lr	9e-5
	hidden-dim	128
	embedding-dim	128
	max-words	10000
	max-length	1000

Table 2: Several ML parameters and DL hyperparameters of the employed models

Transformer-based Models: This work developed the task solutions in four languages: English, Marathi, Telugu, and Spanish. Thus, four different kinds of transformers were explored, including all-MiniLM-L6-v2 (Wang et al., 2020), telugu-sentence-bert-nli (Deode et al., 2023b), marathi-sentence-similarity-sbert (Joshi et al., 2023) and dccuchile-bert-base-spanish-wwm-uncased (Cañete et al., 2023). MiniLM-L6-v2 is a sentence transformer that maps sentences and paragraphs to a 384-dimensional dense vector space and can be used for tasks like clustering or semantic search. Similarly, telugu-sentence-bert-nli is a TeluguBERT model trained on a large dataset. MahaSBERT(marathi-sentence-similarity-sbert) is also fine-tuned on the STS dataset. Before applying these models, we have used other transformer models such as msmarco-distilbert-cos-v5, all-mpnet-base-v2, and all-MiniLM-L12-v2 for English. For evaluating Marathi, we also used pre-trained models like stsb-xlm-r-multilingual , marathi-roberta, and marathi-sentence-bert-nli. Similarly, we have fine-tuned transformer models like LaBSE and telugu-sentence-similarity-sbert for Tamil. Finally, for Spanish, we also used roberta-bne-sentiment-analysis-es and stsb-xlm-r-multilingual. We called all these models from `Huggingface`[1] sentence transformers library. All models were trained on the task datasets. MahaSBERT and TeluguBERT usually perform well for low-resource languages. Dataloader was used from a torch to prepare the data before passing it to the model. We

[1]https://huggingface.co/sentence-transformers

followed a similar approach to train the corresponding transformer model for all languages. Table 3 demonstrates the hyperparameters used to train transformer-based models.

Models	LR	WD	WS
all-Minilm-L6-v2	9e-5	9e-2	750
all-Minilm-L12-v2	9e-5	9e-2	750
marathi–sbert	9e-5	5e-2	500
telugu–bert-nli	9.5e-5	5e-5	750
bert-base-spanish	9e-5	9.5e-7	700

Table 3: Tuned hyperparameter for the transformer-based models, where LR, WD, and WS denotes learning rate, weight decay and warmup steps, respectively

5 Experiments

During the development, this study utilized Python 3 (3.10.12) and Python-based packages from `PyTorch`[2] framework to implement sentence transformers (MiniLM-L6, MarathiSBERT, TeluguSBERT, SpanishSBERT). To implement the models, 29GB of RAM, 16GB of VRAM, and 73.1GB of storage space were used. We utilized NVIDIA Tesla P100 GPU from `Kaggle`[3]. We used pandas (2.1.4) and numpy (1.24.3) to analyze and prepare the data. The ML models were developed with the scikit-learn (1.2.2) packages, and the DL models were trained with Keras (2.13.1) and TensorFlow (2.13.0). The PyTorch (2.0.0) packages, transformers (4.36.2), and sentence transformers (2,6,1) were used to implement transformer models.

The superiority of the models is determined based on the Spearman rank correlation coefficient (ρ) (Sennrich et al., 2015), which measures how well the system predicted rankings of test instances. This work also measures the Kendall correlation (τ) and Pearson correlation (R).

6 Results and Analysis

Table 4 exhibits the evaluation results of ML, DL, and transformer-based models for four languages: English, Marathi, Telegu, and Spanish.

The results demonstrate that the ML models perform poorly. DL models are slightly better, but they need to be better. Transformer-based models demonstrated exceptional performance across all languages. For the English language, Mpnet-v2, Distilbert, and MiniLM-L12 scored 0.821, 0.821, and 0.815, respectively. The top-performing model for English was MiniLM-L6, with a maximum score of 0.822. For the Marathi language, Stsb-xlm, Marathi-BERT-nli and MarathiRoberta received scores of 0.764, 0.866, and 0.810, respectively. The MarathiSBERT model performed best, with a Spearman correlation score of 0.870. For the Telugu language, Telugu-SBERT and LaBSE had scores of 0.761 and 0.804, respectively. The best model for this language is Telugu-BERT-nli, which has a Spearman correlation rank of 0.804. For Spanish, Roberta-bne and Stsb-xlm have scores of 0.659 and 0.677, respectively, but Spanish-BERT outperforms both by 0.677. Figure 3 illustrates the summary of the best-performed models in four languages of the task.

Transformer-based outperformed other models

Language	Models	ρ	τ	R
English	SVR	0.161	0.105	0.181
	RF	0.177	0.115	0.153
	Bagging + RF	0.178	0.114	0.156
	SNN	0.418	0.284	0.473
	MiniLM-L12	0.815	0.614	0.821
	Mpnet-v2	0.821	0.620	0.832
	Distilbert	0.821	0.619	0.829
	MiniLM-L6	**0.822**	0.620	0.832
Marathi	Stsb-xlm	0.764	0.566	0.779
	Marathi-Roberta	0.810	0.619	0.810
	Marathi-BERT-nli	0.866	0.684	0.866
	Marathi-SBERT	**0.870**	0.688	0.875
Telegu	Telugu-SBERT	0.761	0.567	0.795
	LaBSE	0.804	0.608	0.814
	Telugu-BERT-nli	**0.820**	0.617	0.827
Spanish	Roberta-bne	0.659	0.479	0.713
	Stsb-xlm	0.655	0.473	0.707
	Spanish-BERT	**0.677**	0.503	0.719

Table 4: Performance of the employed models on the test set

[2]https://pytorch.org/

[3]https://www.kaggle.com/

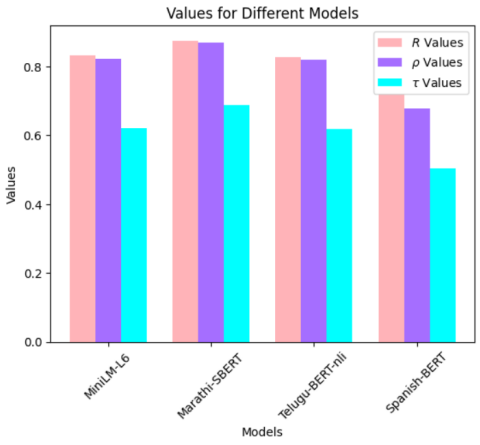


Figure 3: Performance summary of the best model in each task language

by a wide margin for four languages in the task. One explanation is its ability to capture information's bidirectional context. In addition, it produces contextual word representations that make polysemy understandable and enables capturing minute variations in meaning depending on context. Since these transformer-based model has been pre-trained on various linguistic tasks using a sizable dataset, fine-tuning these models improves results.

6.1 Error Analysis

Transformer models can recognize sentence patterns more apparently with better textual features. Thus, these models outperformed ML and DL techniques. Figure 4 depicts some example scores (actual and predicted) regarding two sentences in task languages.

Based on annotations using Best-Worst Scaling, actual scores were computed by deducting the number of times a phrase pair was selected as the least related from the fraction of times it was selected as the most related. The predicted and actual scores are incredibly close in samples 1, 2, and 4. However, the predicted score is more significant for sample 3. This may happen when a sentence is shorter than the longer sentence and contains similar terms. In this case, the model has a more challenging time figuring out the Spearman correlation between these two uneven-length sentences.

7 Conclusion

This study explores the efficacy of various machine learning (ML), deep learning (DL), and transformer-based models for analyzing semantic relatedness within texts across four languages: English, Spanish, Telugu, and Marathi. Experimental assessments reveal subpar performance of both ML and DL models across all languages. However, transformer-based models exhibit superior capabilities in discerning semantic relatedness within the given task. Specifically, the MiniLM-L6 model excels for English, MarathiSBERT for Marathi, TeluguSBERT for Telugu, and SpanishSBERT for Spanish, achieving peak ρ scores of 0.822, 0.870, 0.820, and 0.677, respectively. The study suggests that augmenting training data could enhance the performance of current models. Additionally, leveraging advanced techniques such as Large Language Models (LLM) and Generative Pre-trained Transformers (GPT) holds promise for further improvement.

Sentence #1	Sentence #2	AS	PS
Egypt's Brotherhood stands ground after killings	Egypt: Muslim Brotherhood Stands Behind Morsi	0.7	0.69
Los menonitas amish conascendencia suiza de Galicia se establecieronen 1815 cercade Dubno.	Los amonios menonitas de origen suizo de Galicia se establecieron cerca de Dubno en 1815.	0.8	0.82
మాస్ మహారాజ సినిమాలో మరో ప్రముఖ నటుడు.	మాస్ మహారాజా రవితేజ ఈ వేసవిలో 'క్రాక్' సినిమాతో సందడి చేయాలనుకుంటే తన స్నేహుకు కోవిడ్ 19 బ్రేకులేసింది.	0.38	0.52
सोनूसोबत ज्युनियर कलकत्ता नावाचा बुकीही तेव्हा रॅकेटमध्ये होता.	सोनू आणि अरबाज यांच्यात तेव्हा मोठ्या प्रमाणात पैशांचा व्यवहार झाला.	0.54	0.56

Figure 4: Sample prediction with Similarity scores: Actual (AS) and Predicted (PS)

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on inter-

pretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity.* sem 2012: The first joint conference on lexical and computational semantics—. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), Montréal, QC, Canada*, pages 7–8.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2023. Spanish pre-trained bert model and evaluation data. *arXiv preprint arXiv:2308.02976*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation.

Yiming Chen, Yan Zhang, Bin Wang, Zuozhu Liu, and Haizhou Li. 2022. Generate, discriminate and contrast: A semi-supervised sentence representation learning framework.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023a. L3cube-indicsbert: A simple approach for learning cross-lingual sentence representations using multilingual bert. *arXiv preprint arXiv:2304.11434*.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023b. L3cube-indicsbert: A simple approach for learning cross-lingual sentence representations using multilingual bert. *arXiv preprint arXiv:2304.11434*.

Joseph Gatto, Omar Sharif, Parker Seegmiller, Philip Bohlman, and Sarah Masud Preum. 2023. Text encoders lack knowledge: Leveraging generative llms for domain-specific semantic textual similarity.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Ali Muttaleb Hasan, Noorhuzaimi Mohd Noor, Taha H Rassem, and Ahmed Muttaleb Hasan. 2020. Knowledge-based semantic relatedness measure using semantic features. *International Journal*, 9(2).

Ananya Joshi, Aditi Kajale, Janhavi Gadre, Samruddhi Deode, and Raviraj Joshi. 2023. L3cube-mahasbert and hindsbert: Sentence bert models and benchmarking bert sentence representations for hindi and marathi. In *Science and Information Conference*, pages 1184–1199. Springer.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Martyna Śpiewak, Piotr Sobecki, and Daniel Karaś. 2017. Opi-jsa at semeval-2017 task 1: Application of ensemble learning for computing semantic textual

similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 139–143.

Tokunaga Takenobu. 1994. Text categorization based on weighted inverse document frequency. *Information Processing Society of Japan, SIGNL*, 94(100):33–40.

Xin Tang, Shanbo Cheng, Loc Do, Zhiyu Min, Feng Ji, Heng Yu, Ji Zhang, and Haiqin Chen. 2018. Improving multilingual semantic textual similarity with shared sentence encoder for low-resource languages. *arXiv preprint arXiv:1810.08740*.

Dhruv Verma and SN Muralikrishna. 2020. Semantic similarity between short paragraphs using deep learning. In *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–5. IEEE.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Hao Wu, He-Yan Huang, Ping Jian, Yuhang Guo, and Chao Su. 2017. Bit at semeval-2017 task 1: Using semantic information space to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 77–84.

Xi Yang, Xing He, Hansi Zhang, Yinghan Ma, Jiang Bian, Yonghui Wu, et al. 2020. Measurement of semantic textual similarity in clinical texts: comparison of transformer-based models. *JMIR medical informatics*, 8(11):e19735.

IASBS at SemEval-2024 Task 10: Delving into Emotion Discovery and Reasoning in Code-Mixed Conversations

Mehrzad Tareh and **Aydin Mohandesi** and **Ebrahim Ansari**
Institute for Advanced Studies in Basic Sciences (IASBS)
{m.tareh, mohandesi, ansari}@iasbs.ac.ir

Abstract

In this study, we introduce the SemEval 2024 Task 10, entitled "Emotion Discovery and Reasoning its Flip in Conversation (EDiReF)". Our research presents a comprehensive framework aimed at analyzing emotional dynamics within both Hindi-English mixed-language and English conversations. We extend beyond traditional emotion identification to uncover the triggers behind shifts in emotional states using advanced Natural Language Processing (NLP) techniques. Employing a systematic methodology that encompasses data preprocessing, feature engineering, and the deployment of language models such as GPT-4 and DistilBERT, we unravel the complex interplay of emotions in communication. Our approach yields significant insights, enhancing applications from social media analytics to mental health, thus marking a notable advancement in the integration of emotional intelligence into AI. Noteworthy is our system's achievement of third place on the leaderboard, demonstrating robust performance with a weighted F1-Score of 0.70. This study not only contributes to the field of emotional AI but also paves the way for future research on the nuanced understanding of emotion in mixed-language communications.

1 Introduction

In the age of global digital communication, the English language, with its pervasive influence, has led to a notable increase in bilingual or code-mixed conversations, especially on various social media and messaging platforms. Among these, Hindi-English code-mixing, or "Hinglish", has emerged as a prominent linguistic phenomenon in the Indian subcontinent, embodying the cultural and linguistic interplay of an increasingly globalized society. Despite the widespread occurrence of code-mixed communication, a gap remains in the research landscape, particularly in understanding the emotional dynamics of such interactions (Ramalingam et al., 2023; Attri et al., 2020). Studies in NLP are progressively delving into the significance of emotions in human dialogues, offering promising applications across various domains including human-computer interaction (Kulkarni and Varade, 2023), social media scrutiny (Sharma et al., 2020), and healthcare (Takale, 2024). The EDiReF plays a significant role in this area by scrutinizing emotional expression and shifts in both bilingual (Hindi-English) and monolingual (English) conversations.

Emotion recognition, a facet of affective computing, endeavors to decipher human emotions utilizing diverse technological means. Its potential implications reach far beyond traditional industries, spanning fields as diverse as transportation, finance, and entertainment, promising to revolutionize the way we interact with technology in our daily lives (Guo et al., 2024). Yet, conventional methodologies predominantly focus on analyzing monolingual, single-sentence data, while the EDiReF initiative broadens this horizon to encompass mixed-language dialogues and shifts in emotion within conversations. This acknowledgment of the intricate nature of emotional expression within diverse linguistic and cultural frameworks sets EDiReF apart.

The EDiReF (Kumar et al., 2024a) initiative consolidates three distinct subtasks: Emotion Recognition in Conversation (ERC) in Hindi-English code-mixed conversations (Kumar et al., 2023), Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations, and EFR in English conversations (Kumar et al., 2022; Kumar et al., 2024b). Each subtask presents its own set of challenges in comprehending and interpreting emotions within conversational contexts. In the ERC subtask, we were tasked with assigning emotion labels to each utterance within a dialogue, drawing from a predefined spectrum of emotions. This demands algorithms capable of discerning and distinguishing subtle emotional cues embedded within mixed-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1229–1238

language exchanges, thereby facilitating a deeper comprehension of emotional dynamics in bilingual interactions.

The EDiReF shared task serves as a crucial avenue for researchers to delve into innovative methodologies, exchange insights, and benchmark their models using real-world conversational data. This initiative contributes significantly to the overarching objective of advancing affective computing and deepening our comprehension of human emotions within natural language interactions.

2 Related Work

The exploration of emotion in human dialogue, especially within the realm of code-mixed conversations, represents a burgeoning field of study that intersects with computational linguistics, affective computing, and cross-cultural communication. This section reviews seminal works and recent advancements that set the stage for our investigation into EDiReF.

Emotion recognition, an integral part of the expanding domain of affective computing, endeavors to decipher and interpret human emotions through technological means. This interdisciplinary field amalgamates aspects of computer science, psychology, and neuroscience to forge innovative devices capable of recognizing, understanding, and reacting to human emotional states (Montag and Davis, 2020). Such advancements hold the potential to significantly transform human-computer interactions, promising to enhance user experiences across various sectors such as retail, finance, and entertainment, thereby enabling personalized and intuitive interactions (Matin and Valles, 2020).

Historically, the focus has predominantly been on monolingual, single-sentence analyses; however, the EDiReF task expands this horizon by exploring mixed-language dialogues and the dynamics of emotion flips within conversations. This forward-looking approach acknowledges the intricacies of communication, emphasizing the significance of context, cultural differences, and linguistic diversity in the accurate interpretation of emotions. By incorporating mixed-language data, the task addresses the growing occurrence of bilingual conversations in global communications, aiming to develop more inclusive and precise emotion detection algorithms that reflect the true complexity of human interactions (Muhammadiyeva, 2022).

The availability and quality of annotated datasets for training and evaluation emerge as significant hurdles, especially for less represented languages. The quest for consistency in annotations across languages and emotions further adds to the complexity (Garg, 2020). Furthermore, the requirement for effective cross-lingual representation learning highlights the need for models to accurately capture language-specific features and emotions, necessitating sophisticated approaches in transfer learning (Ranaldi and Pucci, 2023). Additionally, identifying trigger utterances for emotion flips introduces another layer of complexity, requiring a nuanced understanding of dialogue dynamics and contextual cues (Kumar et al., 2024c). The scalability and generalization of models across different conversational contexts, languages, and domains remain formidable challenges.

In summary, while existing research provides valuable insights into emotion recognition and code-mixing, there remains a notable paucity of studies specifically addressing the dynamics of emotion in code-mixed conversations. Our work seeks to fill this gap, leveraging state-of-the-art NLP techniques to analyze emotional content and reasoning in Hindi-English mixed-language dialogues. By doing so, we contribute to the broader discourse on advancing affective computing in multilingual and multicultural contexts.

3 Task Description

This task comprises three subtasks, each addressing distinct aspects of emotional understanding and analysis in dialogues.

3.1 Emotion Recognition in Conversations (ERC)

Emotion Recognition in Conversations stands at the forefront of computational linguistics and artificial intelligence, employing advanced algorithms and methodologies to decipher emotional nuances within textual exchanges. By meticulously collecting and preprocessing data, ERC systems utilize machine learning models to classify emotional states expressed in conversations, ranging from joy to sadness and anger to fear (Dessai and Virani, 2022). Such advancements hold transformative potential across diverse sectors, from revolutionizing customer service interactions to enhancing mental health support through an early intervention based on text analysis. The applications of ERC extend beyond sentiment analysis, playing pivotal

roles in education, market research, and human-computer interaction (Loveland, 2011). Moreover, ERC serves as a potent tool for market research, enabling companies to gauge public sentiment towards products or brands by analyzing social media conversations and consumer reviews (Razouk et al., 2023).

Emotion	Proportion(%)
Neutral	45.4
Joy	19.0
Anger	9.4
Sadness	7.3
Fear	6.3
Contempt	6.1
Surprise	4.9
Disgust	1.4

Table 1: Distribution of emotions in dataset for ERC

Despite the immense potential of ERC, ethical considerations regarding privacy, biases in emotion detection algorithms, and responsible data handling are paramount. However, as ERC continues to evolve, particularly in navigating the complexities of Hindi-English code-mixed conversations, it offers promise in bridging cultural divides and enabling more nuanced sentiment analysis in multicultural settings. Nevertheless, addressing ethical concerns through responsible research and implementation is crucial to fostering a more inclusive and empathetic digital landscape (Bagora et al., 2022; Sitaram et al., 2015).

3.2 Emotion Flip Reasoning (EFR)

Dataset	0.0(%)	1.0(%)
Subtask 2	93.5	6.5
Subtask 3	84.7	15.3

Table 2: Distribution of triggers in dataset for EFR

3.3 Code-mixing in Conversations

Code-mixing in conversations occurs when speakers blend elements from two or more languages within the same discourse. This phenomenon is prevalent in multilingual communities where individuals are fluent in multiple languages and switch between them based on social context, familiarity, or communicative needs. In such conversations, speakers may switch between languages mid-sentence or incorporate phrases, expressions, or even entire sentences from one language into another. Code-mixing adds richness and depth to communication, allowing speakers to draw from a wider linguistic repertoire to express their thoughts and convey nuances that may not be readily available in a single language. Hindi-English code-mixing, commonly known as Hinglish, is a prominent example of code-mixing in conversations, especially in regions with significant bilingual populations like India (Kodali et al., 2022). In Hinglish conversations, speakers seamlessly integrate Hindi and English elements, creating a unique linguistic fusion that reflects the cultural and linguistic diversity of the Indian subcontinent. Hinglish code-mixing serves as a linguistic bridge, allowing speakers to navigate between their cultural identities and accommodate the diverse linguistic backgrounds of their interlocutors (Jawahar et al., 2021).

The use of code-mixing, whether in general conversations or specifically in Hinglish, serves several communicative functions. Firstly, it facilitates smoother communication by allowing speakers to express themselves using the most appropriate linguistic resources available to them. Additionally, code-mixing can convey social and cultural affiliations, signaling aspects of the speaker's identity such as ethnicity, education level, or social status. Moreover, code-mixing can serve pragmatic functions, such as clarifying meanings, emphasizing certain points, or creating humorous effects (Vogh, 2022).

3.4 Hindi-English Code-mixed Conversations

Code-mixed conversations, blending Hindi and English, are a common phenomenon in bilingual societies like India. This linguistic fusion reflects the cultural and social dynamics of the populace. In everyday interactions, individuals effortlessly switch between the two languages, often using Hindi for informal contexts and English for formal or technical discussions. These fluid exchanges showcase the flexibility and richness of language usage in diverse settings (Yadav et al., 2020; Mukherjee, 2019).

Code-mixing isn't just about linguistic versatility; it's also deeply ingrained in identity expression. By integrating Hindi and English, speakers navigate their cultural affiliations and social environments. This blending of languages is not only a means of communication but also a reflection of one's hybrid cultural identity (Attri et al., 2020). Furthermore, code-mixed conversations play a cru-

cial role in digital communication, especially on social media platforms and messaging apps. In the virtual realm, users often employ a mix of Hindi and English to cater to a wider audience while maintaining a sense of familiarity and belonging. This phenomenon has led to the emergence of unique online subcultures and linguistic trends. In essence, code-mixed conversations not only bridge linguistic divides but also serve as a vibrant expression of cultural fusion in a globalized world (Dabrowska, 2019).

Task	Hindi(%)	English(%)
Subtask 1	59.8	40.2
Subtask 2	39.8	60.2
Subtask 3	17.7	82.3

Table 3: Hindi vs English proportions in dataset

4 Dataset Description

The organizers have supplied and divided the datasets for participants into train, development, and test sets. The table 4 provided contains details regarding the number of instances used for training, development, and testing.

Dataset	Train	Dev	Test
Subtask 1	8,506	1,354	1,580
Subtask 2	98,777	7,462	7,690
Subtask 3	35,000	3,522	8,642

Table 4: Summary of dataset split

4.1 Introduction to MaSac Dataset

The MaSaC dataset is a carefully curated collection designed to investigate code-mixed dialogue interactions in the Indian context, drawing from the popular television series "Sarabhai v/s Sarabhai." This dataset captures the authentic dynamics of conversations in a multi-party, multimodal setting, predominantly featuring a blend of Hindi and English languages. With over 11,000 utterances dedicated to task 1 and 114,000 to task 2, it provides researchers with a rich corpus to explore various aspects of language usage and communication dynamics. Each utterance in a dialogue has been labeled by any of these eight emotions: Anger, Disgust, Sadness, Joy, Neutral, Surprise, Fear, and Contempt. Its multimodal nature, incorporating textual, auditory, and visual elements from the television show, offers a comprehensive view of dialogue interactions, while its labeling of emotions for each utterance enriches understanding of the emotional nuances within the conversations (Kumar et al., 2023).

Furthermore, the MaSaC dataset holds significant promise for diverse research endeavors. In the realm of natural language understanding, it facilitates the development and evaluation of models capable of comprehending and generating code-mixed utterances, thereby enhancing language processing capabilities in multilingual environments. Additionally, computational linguistics, enables investigations into code-switching phenomena and sociolinguistic variations, shedding light on language usage patterns and communicative strategies. Beyond academia, the dataset's exploration of socio-cultural aspects embedded within language interactions offers valuable insights for sociolinguists and cultural researchers, fostering a deeper understanding of identity expression, social dynamics, and cultural nuances depicted in televised narratives (Chakraborty, 2021).

4.2 Introduction to MELD Dataset

The Multimodal Emotion Lines Dataset (MELD) stands out as a pivotal resource for researchers delving into the intricate realm of emotion recognition, particularly within the context of multiparty conversations. Comprising over 47,000 utterances extracted from the beloved television series Friends, MELD is an extension and enhancement of EmotionLines (Chen et al., 2018), and presents an extensive collection of genuine exchanges, showcasing diverse interactions among numerous speakers engaged in dynamic dialogues. Unlike its predecessor, MELD adopts a multimodal approach by incorporating textual transcripts.

However, it is crucial to note that the dataset provided by the organizers did not include audio-visual cues, focusing instead on the textual aspect to understand emotional communication. Each conversation in the dataset is carefully annotated, presenting detailed labels for emotions expressed, making it a valuable resource for supervised learning techniques. The dataset labels each utterance with one of seven emotions: Anger, Disgust, Sadness, Joy, Neutral, Surprise, and Fear, aiming to capture the full spectrum of emotional dynamics in conversation (Kumar et al., 2024c; Kumar, 2023).

5 Experimental Setup

This section outlines the various aspects of the experimental setup, including dataset preparation, evaluation metrics, baseline systems, and training methodologies.

5.1 JSON Parsing

In our exploration of this task, we encounter datasets structured in JSON format, encapsulating dialogues among multiple speakers annotated with emotions and triggers. To effectively manage this data, we employ a detailed parsing process. Initially, we load the JSON data into memory, followed by iterative processing of each dialogue entry. We extract pertinent details such as episode ID, speaker ID, emotion label, trigger label, and utterance text, organizing them into a structured DataFrame format. Additionally, to ensure each entry's unique identification, we generate non-negative integer IDs for each dialogue instance, facilitating seamless referencing during subsequent data analysis and model development stages.

By effectively extracting essential information and generating unique IDs for dataset entries, we can navigate through the data with ease, enabling profound understanding and fostering advancements in emotion recognition and reasoning within code-mixed and multi-party conversation dialogues.

5.2 Dataset Preprocessing

We meticulously executed preprocessing steps on the dataset to guarantee uniformity of data and streamline subsequent analysis and model development. Initially organized by task organizers (refer to table 4), the dataset underwent thorough text normalization techniques including lowercase conversion and removal of redundant characters and excessive punctuation to enhance readability and consistency. Subsequent tokenization segmented the text for deeper analysis, followed by language-specific procedures such as stopword removal, lemmatization, and stemming to refine textual content. Language identification techniques were also employed to differentiate between Hindi and English segments, allowing for targeted preprocessing steps. Our collaborative efforts standardized and formatted the dataset in alignment with the prescribed framework of the shared task, laying a robust foundation for effective participation and further analysis.

5.2.1 Dataset Cleaning and Standardization

- Addressed data integrity by handling less than 5 samples with invalid values, assigning the most common value within the dataset to maintain consistency.
- Standardized speakers' names to ensure uniformity by resolving discrepancies like varying capitalization.
- Enhanced dataset structure by assigning non-negative IDs to episodes and speakers, enabling more efficient data processing.
- In the EFR task, focused solely on triggers with a value of 0.0, resulting in an emotional inversion where the trigger value shifted to 1.0.

5.3 Dataset Translation

Initially, while relying on Google Translate, our comprehensive manual verification process unveiled discrepancies and inaccuracies, compelling us to explore alternatives of greater reliability and precision. In this context, GPT-4 (OpenAI, 2022) emerged as a pivotal tool, distinguished for its contextual comprehension and translation proficiency. Our translation methodology was meticulously crafted to prioritize fidelity to the original dialogues, thereby ensuring the preservation of nuanced semantic and syntactic elements across linguistic boundaries. By harnessing the capabilities of GPT-4 in conjunction with supplementary support from Google Translate, we embarked on a systematic translation endeavor aimed at capturing the inherent complexity and subtlety of the conversations. The outcomes were noteworthy, as GPT-4 consistently exceeded expectations, exhibiting an exceptional aptitude for encapsulating the intricate nuances of emotional expression and linguistic subtleties. Armed with these carefully refined translations, we deftly integrated them into the final English datasets, confident in their accuracy and faithfulness to the original discourse. This meticulous approach not only ensured linguistic coherence but also paid homage to the rich cultural nuances embedded within the conversations (Nakayama et al., 2019).

5.3.1 Dataset Normalization

Hindi-English code-mixed conversations posed a significant challenge, especially with restrictions on available tools due to sanctions. Our initial

attempts with a robot browser yielded unsatisfactory results, prompting us to leverage the GPT-4 API in conjunction with the spaCy. This combination significantly outperformed traditional translation services, like Google Translate, in accuracy. Post-translation, we employed a pre-trained model (Kunchukuttan, 2020) to normalize utterances to standard Hindi/Romani, followed by tokenization and analysis with Morph to adapt the analyzed form.

5.3.2 Feature Engineering

For sentiment analysis, we utilized the INT8 DistilBERT model (He and Wenz, 2022) through the Cloudflare API, streamlining the process to calculate positive and negative scores for each utterance. A novel feature introduced was the calculation of polarity differences between consecutive utterances within an episode, aiding in the detection of emotion flips.

5.4 Dataset Polarity

In the Dataset Polarity section, we present the essential analysis conducted on the provided dataset. Here, we outline the methodology we employed to evaluate the polarity of utterances within the dataset. Utilizing cutting-edge natural language processing technology, specifically DistilBERT from Intel, we undertook the task of calculating polarity scores for each utterance.

Polarity scores serve as a quantitative measure of the sentiment conveyed within individual utterances. Our approach involved leveraging DistilBERT's pre-trained language understanding capabilities to discern the underlying sentiment expressed in the text. By employing this state-of-the-art model, we aimed to capture subtle emotional undertones present in the conversations, particularly in the context of Hinglish dialogues. Furthermore, to enhance our understanding of the dataset, we computed the difference in polarity scores between utterances. This differential analysis provides insights into the shifts in sentiment within conversations, a crucial aspect for tasks such as ERC and EFR. By discerning fluctuations in sentiment, we gain valuable information for identifying trigger utterances for emotion flips in multi-party dialogues.

The integration of DistilBERT from Intel in our polarity analysis underscores our commitment to leveraging state-of-the-art techniques for robust sentiment analysis. Through this punctilious approach, we aim to provide a comprehensive understanding of the emotional dynamics inherent in the dataset, thereby facilitating advancements in emotion recognition and reasoning tasks within code-mixed conversations.

6 Methodology

To explore the intricacies of ERC and EFR within Hindi-English code-mixed conversations, as well as EFR in English conversations, our methodology integrates a blend of traditional machine learning models with the cutting-edge capabilities of transformer-based architectures.

At the outset, we harness the advanced linguistic comprehension of GPT-4, a state-of-the-art language model, to ascertain the emotions embedded in each utterance of the dialogue. The prowess of GPT-4 lies in its nuanced grasp of context and the subtleties of natural language, rendering it highly effective for the preliminary prediction of emotions within conversations. Subsequently, we employ a spectrum of classical machine learning techniques, including Random Forest, Support Vector Machines (SVM), Logistic Regression, and Naive Bayes classifiers. These algorithms are foundational to the field of natural language processing and serve as a benchmark for evaluating the advanced methodologies employed later. This hybrid modeling approach aims to capitalize on the depth and context-awareness provided by transformer-based models, like GPT-4, while also valuing the interpretability and established nature of classical machine learning techniques. By leveraging this diverse array of models, our objective is to harness the strengths of both modern and traditional approaches to optimize performance across the ERC and EFR tasks in conversations conducted in both Hinglish and English.

For the task of Emotion Recognition in Conversation, we adopt the weighted F1-score as our primary evaluation metric (see Table 7 and 8). This metric is chosen for its ability to provide a balanced measure of the model's precision and recall, while also accounting for class imbalances that are common in real-world datasets. This nuanced evaluation allows us to assess the model's ability to accurately recognize emotions across a diverse set of conversations. In the case of Emotion Flip Reasoning, our focus shifts toward precision as the primary evaluation metric (see table 6). Precision is particularly relevant for EFR tasks as it measures the model's accuracy in identifying the specific in-

stances where an emotional flip occurs within the conversation. This metric enables us to refine the model's performance in pinpointing these critical junctures, thereby ensuring high reliability in the model's reasoning capabilities.

By employing this diverse range of models, we aimed to leverage both the sophistication of transformer-based architectures and the interpretability of classical machine learning algorithms. This hybrid approach allowed us to explore different facets of the data and optimize performance across the ERC and EFR tasks in both Hinglish and English conversations.

7 Results and Discussions

In our study, we present a comprehensive analysis of our findings, which is predicated on our noteworthy accomplishment of securing the third position on the Codalab leaderboard among forty participants.

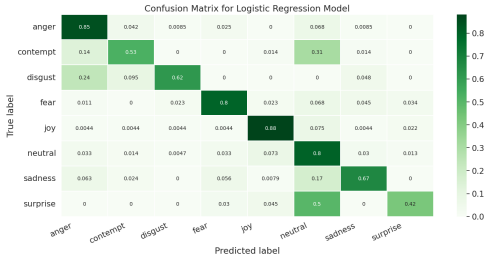
Team	Subtask 1	Subtask 2	Subtask 3
MasonTigers	0.78	0.79	0.79
Knowdee	0.73	0.66	0.61
IASBS	**0.70**	**0.12**	**0.25**
Alden_Jenish	0.66	0.07	0.04

Table 5: Top four participants' scores on CodaLab

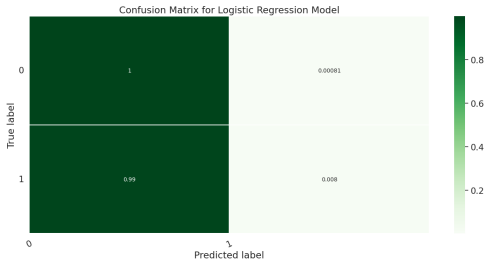
This achievement underscores the efficacy of our comprehensive framework, which integrates advanced language models like GPT-4 and DistilBERT, alongside sophisticated data preprocessing and feature engineering techniques, to explore the nuanced interplay of emotions in conversations.

In our investigation, the evaluation of various classification models across the three subtasks of emotion recognition and reasoning exhibited distinct performance characteristics, as detailed in Tables 6, 7 and 8. For ERC, the models demonstrated a competitive range of F1-scores, with GaussianNB slightly outperforming others in terms of precision and recall, reflecting a balanced capacity for emotion classification within dialogues. The nuanced demands of EFR in both Hinglish and English conversations required precision as a critical metric due to the importance of accurately identifying emotional shifts. In this context, the SVM model showed notable efficacy, particularly in English conversations, as it achieved a precision of 0.79, indicating a strong ability to discern the nuanced triggers of emotion flips. Further analysis in Table 7 revealed that for EFR within Hinglish code-mixed conversations, the Logistic Regression model surfaced as a frontrunner, achieving a precision of 0.74, underscoring its capability to handle the intricacies of code-switching and the emotional dynamics inherent in bilingual discourse.

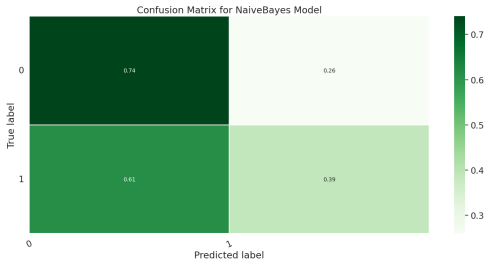
Meanwhile, Table 8 underscores the adaptability of machine learning models to the linguistic complexity and emotional subtleties in English conversations, with the Random Forest model marking a precision of 0.81, reflecting its robustness in parsing and understanding the nuanced indicators of emotional transitions.



(a) Confusion matrix of task 1



(b) Confusion matrix of task 2



(c) Confusion matrix of task 3

Figure 1: Confusion matrix of all three subtasks

These findings not only elucidate the strengths and limitations inherent in various computational models' ability to comprehend the complexities of emotional nuances within conversational contexts but also underscore the imperative for further refinement. It is crucial to enhance the sensitivity and

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.77**	**0.77**	**0.77**	**0.77**
SVM	0.77	0.77	0.77	0.77
GaussianNB	0.78	0.76	0.76	0.76
MLP	0.76	0.76	0.76	0.76
LDA	0.77	0.76	0.76	0.76
KNN	0.75	0.75	0.75	0.75
Random Forest	0.76	0.76	0.74	0.76
AdaBoost	0.75	0.75	0.74	0.75
QDA	0.62	0.71	0.66	0.71

Table 6: Comparative Evaluation Results of Various Classification Models for ERC (Hinglish)

Model	Precision	Recall	F1-Score	Accuracy
SVM	**0.79**	**0.89**	**0.83**	**0.98**
LDA	0.79	0.89	0.83	0.89
AdaBoost	0.79	0.89	0.83	0.89
Logistic Regression	0.8	0.77	0.78	0.77
KNN	0.8	0.73	0.76	0.73
Random Forest	0.81	0.72	0.76	0.72
BernouliNB	0.82	0.7	0.75	0.71
QDA	0.81	0.71	0.75	0.71
MLP	0.81	0.69	0.74	0.69

Table 7: Comparative Evaluation Results of Various Classification Models for EFR (Hinglish)

Model	Precision	Recall	F1-Score	Accuracy
SVM	**0.79**	**0.89**	**0.83**	**0.98**
LDA	0.79	0.89	0.83	0.89
AdaBoost	0.79	0.89	0.83	0.89
Logistic Regression	0.8	0.77	0.78	0.77
KNN	0.8	0.73	0.76	0.73
Random Forest	0.81	0.72	0.76	0.72
BernouliNB	0.82	0.7	0.75	0.71
QDA	0.81	0.71	0.75	0.71
MLP	0.81	0.69	0.74	0.69

Table 8: Comparative Evaluation Results of Various Classification Models for EFR (English)

precision of these models, especially within the intricate landscape of multilingual and multicultural communication.

8 Conclusion

Our study delves into how language and culture intertwine with emotional dynamics in bilingual conversations, revealing new insights through an examination of Emotion Recognition in Conversations and Emotion Flip Reasoning. It highlights the need for computational models to accurately interpret emotional nuances across different cultures and languages, advocating for interdisciplinary efforts to enhance AI's empathy and cultural awareness. The research aims to improve global understanding and connectivity, contributing to better human-computer interaction and societal unity. Future directions include expanding the research to more languages and cultures, integrating sociolinguistic and anthropological insights into computational models, and exploring the role of multimodal communication in emotion recognition to develop more sophisticated AI systems.

9 Future Work

Moving forward, our research trajectory entails several promising avenues for exploration. Firstly, we aim to broaden our linguistic and cultural scope by expanding our investigations to encompass a more extensive array of languages and cultural backgrounds. Additionally, we seek to enrich our computational models by integrating insights from fields such as sociolinguistics and anthropology, thereby fostering a more nuanced understanding of emotional expression within diverse societal frameworks. Moreover, we endeavor to delve into the realm of multimodal communication to unravel the intricate interplay between verbal and nonverbal cues in emotion recognition. By embracing these future directions, we aspire to cultivate more sophisticated AI systems capable of seamlessly navigating the intricate tapestry of human emotion across diverse cultural landscapes.

References

Shree Harsh Attri, T.V. Prasad, and G. Ramakrishna. 2020. Hiphet: A hybrid approach to translate code mixed language (hinglish) to pure languages (hindi and english). *Computer Science*, 21(3).

Aditi Bagora, Kamal Shrestha, Kaushal Maurya, and Maunendra Sankar Desarkar. 2022. Hostility Detection in Online Hindi-English Code-Mixed Conversations. In *14th ACM Web Science Conference 2022*. ACM.

Tanmoy Chakraborty. 2021. Multi-modal Sarcasm Detection and Humor Classification in Code-mixed Conversations. *ArXiv*, abs/2105.09984.

Béatrice Chen, Hui Fang, Lun-Wei Ku, Hsin-Yang Li, Chao-Chun Lin, and Hung-Yu Lee. 2018. Emotionlines: An Emotion Corpus of Multi-Party Conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marta Dabrowska. 2019. What is indian in indian english? markers of indianness in hindi-speaking users' social media communication. *Journal on Asian Linguistic Anthropology*.

Amita Umesh Dessai and Hassanali G. Virani. 2022. *Emotion Detection and Classification Using Machine Learning Techniques*. IGI Global.

Neha Garg. 2020. Annotated Corpus Creation for Sentiment Analysis in Code-mixed Hindi-English (Hinglish) Social Network Data. *Indian Journal of Science and Technology*, 13(40).

R. Guo, H. Guo, L. Wang, et al. 2024. Development and Application of Emotion Recognition Technology — A Systematic Literature Review. *BMC Psychology*.

Xin He and Yu Wenz. 2022. distilbert-base-uncased-finetuned-sst-2-english-int8-static.

Ganesh Jawahar, El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, and Laks V.S. Lakshmanan. 2021. Exploring Text-to-Text Transformers for English to Hinglish Machine Translation with Synthetic Code-Mixing. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics.

Prashant Kodali, Anmol Goel, Monojit Choudhury, Manish Shrivastava, and Ponnurangam Kumaraguru. 2022. SyMCoM - Syntactic Measure of Code Mixing: A Study Of English-Hindi Code-Mixing. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics.

Rahul Kulkarni and Pradip Varade. 2023. NLP and Human-Computer Interaction: Enhancing User Experience Through Language Technology. *International Journal for Research in Applied Science and Engineering Technology*.

Shivani Kumar. 2023. Emotion Flip Reasoning in Multiparty Conversations_supp1-3289937.pdf.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024a. SemEval 2024 – Task 10: Emotion Discovery and Reasoning Its Flip in Conversation (EDiReF). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2024b. Emotion Flip Reasoning in Multiparty Conversations. *IEEE Transactions on Artificial Intelligence*.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2024c. Emotion Flip Reasoning in Multiparty Conversations. *IEEE Transactions on Artificial Intelligence*.

Shivani Kumar, Ramaneswaran S, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. From Multilingual Complexity to Emotional Clarity: Leveraging Commonsense to Unveil Emotions in Code-Mixed Dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering Emotion and Reasoning Its Flip in Multi-Party Conversations Using Masked Memory Network and Transformer. *Knowledge-Based Systems*.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Susan Loveland. 2011. Human Computer Interaction That Reaches Beyond Desktop Applications. In *Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM.

Rezwan Matin and Damian Valles. 2020. A Speech Emotion Recognition Solution-Based on Support Vector Machine for Children with Autism Spectrum Disorder to Help Identify Human Emotions. In *2020 Intermountain Engineering, Technology and Computing (IETC)*. IEEE.

Christian Montag and Kenneth L. Davis. 2020. punctum books. [link].

O. Muhammadiyeva. 2022. Use of Language Elements in the Process of Social Communication. *Zamonaviy*.

Siddhartha Mukherjee. 2019. Deep Learning Technique for Sentiment Analysis of Hindi-English Code-Mixed Text Using Late Fusion of Character and Word Features. In *2019 IEEE 16th India Council International Conference (INDICON)*. IEEE.

Sahoko Nakayama, Takatomo Kano, Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2019. Recognition and Translation of Code-Switching Speech Utterances. In *2019 22nd Conference of the Oriental Chapter of the International Committee for the Coordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA)*. IEEE.

OpenAI. 2022. GPT-4 Technical Report.

Anita Ramalingam, Nirav Agarwal, and Harshvardhan Arvind Singh. 2023. Detection of Virulent Messages Written in Code-Mixed Hindi-English Language. In *Recent Trends in Data Science and its Applications*. River Publishers.

Leonardo Ranaldi and Giulia Pucci. 2023. Does the English Matter? Elicit Cross-lingual Abilities of Large Language Models. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*. Association for Computational Linguistics.

Ayoub Razouk, Hamza Melliani, Jallal Mohamed El Adnani, and Moulay El Mehdi Falloul. 2023. Analyzing Public Sentiment Towards Islamic Finance Through Social Media: Using Sentiment Analysis on Twitter Data. *Revue d'Intelligence Artificielle*, 37.

Dipti Sharma, Munish Sabharwal, Vinay Goyal, and Mohit Vij. 2020. Sentiment Analysis Techniques for Social Media Data: A Review. In *First International Conference on Sustainable Technologies for Computational Intelligence*. Springer.

Dinkar Sitaram, Savitha Murthy, Debraj Ray, Devansh Sharma, and Kashyap Dhar. 2015. Sentiment Analysis of Mixed Language Employing Hindi-English Code Switching. In *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*. IEEE.

Dattatray G. Takale. 2024. A Study of Natural Language Processing in Healthcare Industries. *Journal of Web Applications and Cyber Security*.

Kendall Vogh. 2022. *Chapter 14. Code-Mixing and Semantico-Pragmatic Resources in Francophone Maine: Meanings-in-Use of Yeah/Yes and Ouais/Oui*. John Benjamins Publishing Company.

Konark Yadav, Aashish Lamba, Dhruv Gupta, Ansh Gupta, Purnendu Karmakar, and Sandeep Saini. 2020. Bi-LSTM and Ensemble Based Bilingual Sentiment Analysis for a Code-mixed Hindi-English Social Media Text. In *2020 IEEE 17th India Council International Conference (INDICON)*. IEEE.

Deja Vu at SemEval 2024 Task 9: A Comparative Study of Advanced Language Models for Commonsense Reasoning

Trina Chakraborty
Shahjalal University of Science & Technology
trina41@student.sust.edu

Md. Marufur Rahman
Shahjalal University of Science & Technology
marufurr701@gmail.com

Omar Faruqe Riyad
Shahjalal University of Science & Technology
riyad.omf@gmail.com

Abstract

This research systematically forms an impression of the capabilities of advanced language models in addressing the BRAINTEASER task introduced at SemEval 2024, which is specifically designed to explore the models' proficiency in lateral commonsense reasoning. The task sets forth an array of Sentence and Word Puzzles, carefully crafted to challenge the models with scenarios requiring unconventional thought processes. Our methodology encompasses a holistic approach, incorporating preprocessing of data, fine-tuning of transformer-based language models, and strategic data augmentation to explore the depth and flexibility of each model's understanding. The preliminary results of our analysis are encouraging, highlighting significant potential for advancements in the models' ability to engage in lateral reasoning. Further insights gained from post-competition evaluations suggest scopes for notable enhancements in model performance, emphasizing the continuous evolution of the models in mastering complex reasoning tasks.

1 Introduction

The reasoning ability of the human brain demonstrates a dualistic problem-solving approach, which integrates both vertical and lateral methodologies (Bala, 2014). Vertical thinking places emphasis on a methodical and logical examination, executed in a sequential fashion, guided by established norms and regulations. On the contrary, lateral thinking (De Bono, 1970) promotes innovation and fosters the ability to perceive challenges from distinct, frequently unusual observation points, thereby encouraging individuals to go beyond conventional limitations.

Over the past few years, significant progress has been made in the booming domain of Natural Language Processing (NLP), as novel technologies aim to mimic the the complicated ways in which human think (Kumar et al., 2023; Koivisto and Grassini, 2023). This undertaking overcomes conventional logical reasoning and dives into the domain of creative cognition, wherein machines are engineered to creatively navigate and interpret the complex nature of human language and thought processes. Out of these efforts, the BRAINTEASER task is particularly noteworthy for its pioneering aspect (Jiang et al., 2023). The challenge, which is part of the SemEval 2024, has been carefully constructed to evaluate a model's capacity for lateral thinking and its aptitude for questioning and redefining conventional commonplace assumptions.

BRAINTEASER (Jiang et al., 2024) takes a significant progress in the direction of bridging the divide that exists between the cognitive flexibility of humans and that of machines (Boyacı et al., 2023), exceeding the bounds of study. By selecting puzzles that require perception at both the sentence and word levels, this task highlights the significant technological advancement towards machines capable of creative thinking and logical conclusions that beat the apparent. The task encourages participants overcome the limitations of natural language processing (NLP) models by evaluating their capacity to interpret and decode language in a manner that emulates the creative reasoning of humans. Motivated by the unique characteristics of the assignment and the potential it provides to advance the domain of NLP technology, our group wholeheartedly accepted the challenge presented by BRAINTEASER. Our approach was experimental, leveraging a variety of transformer models to explore their capacity for creative and lateral thinking (Hashim et al., 2023). These models, known for their effectiveness (Nassiri and Akhloufi, 2023) in understanding and generating human language, were put to the test to see if they could indeed mimic the thought processes traditionally attributed to humans. We ranked at 20th in each of the subtasks and in average the rank was 31. The experience was rich with learning opportunities, offering

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1239–1244

us valuable insights into the capabilities and limitations of current technologies when faced with tasks that require a departure from conventional reasoning.

2 Task and Data Description

The task (Jiang et al., 2024) is making a system which is evaluated on understanding sentences and words in ways that defy usual expectations. It has two main challenges:

- Sentence Puzzle: the system must interpret sentences in unexpected ways.
- Word Puzzle: the system need to find unconventional meanings of words.

The task employs specific tests to make sure the system analyzes information deeply instead of merely remembering answers. These tests involve altering the phrasing or setting of questions without changing the basic problem, known as semantic and context reconstruction. The systems are evaluated based on two primary factors: their ability to address single questions, referred to as instance-based performance, and their consistency in answering groups of related questions, known as group-based performance. The goal of this task is to advance the system's abilities in problem-solving and creative thinking

The dataset (Jiang et al., 2023) provided for the task includes two distinct types of files, one for sentence puzzles and another for word puzzles. Each file is rich with essential elements such as the posed question, the correct answer, three alternative options (distractors), labels, a list of choices, and the sequence in which these choices are presented. During the training phase, the dataset comprises 507 sentence puzzles and 396 word puzzles, demonstrating a comprehensive range of scenarios for model training. For the testing phase, the dataset narrows down to 120 sentence puzzles and 96 word puzzles, aimed at rigorously evaluating the models' understanding and reasoning capabilities in both puzzle types

3 System Description

3.1 Data Pre-processing

Our data pre-processing for the BRAINTEASER task involved meticulous steps to prepare the dataset for effective model training. Starting with the loading and merging of two numpy arrays—'SP-train.npy' for sentence puzzles and 'WP-train.npy' for word puzzles—we created a unified dataset comprising a diverse range of puzzles. Recognizing the importance of an unbiased dataset for model training, we employed a two-step randomization process. Initially, we randomized the order of the combined dataset. Subsequently, after converting the dataset into a pandas DataFrame, we applied an additional shuffle to guarantee thorough randomness. Given the difficulties of the puzzles, converting them into a binary classification format presented unique challenges. Each puzzle was transformed into a series of question-choice pairs labeled as correct or incorrect. This binary labeling was crucial for training our models to detect the subtle differences between potential answers, thereby enhancing their reasoning capabilities and language understanding. This careful preparation, including a strategic split of 90% for training and 10% for validation, ensured that our models were ready for the BRAINTEASER challenge.

3.2 Data Augmentation

To enrich the dataset and enhance model robustness, we implemented several data augmentation techniques. These included synthesizing new puzzle questions by paraphrasing existing ones and introducing variations in the dataset to simulate a wider range of linguistic structures and puzzle formats. Such augmentation not only expanded the diversity of our training set but also provided our models with a broader linguistic context to learn from, thereby improving their generalization capabilities. This strategy was particularly beneficial in taking decisions to choose the best model.

3.3 Encoding for Models

In the next step, we take a streamlined approach to improve question-answering models through a custom class, integrating seamlessly with PyTorch's Dataset framework. This class, initialized with essential components like questions, answers, labels, a tokenizer, and a max token length, ensures comprehensive preparation of question answer pairs for training. We have tried to apply encoding each pair to produce a dictionary containing merged question-answer texts, input IDs, attention masks, and labels, all conforming to a specified maximum token length. This process, emphasizing special tokens, padding, and truncation, readies each pair for model training, significantly simplifying data handling. The class is instrumental in converting

raw data into a format conducive to learning, thus enhancing the models' ability to generate insightful responses.

3.4 Model Training

For this part, we considered different transformer models to experiment the performance. The models are :

- BERT (Do and Phan, 2022) (Bidirectional Encoder Representations from Transformers) revolutionized NLP by training on a massive corpus in a bidirectional manner, enabling it to grasp context from both directions, thus providing a deep understanding of language distinction.
- XLNet (Ghavidel et al., 2020) extends upon BERT by employing a permutation-based training method, which allows it to capture the bidirectional context more effectively, making it particularly adept at handling tasks requiring a nuanced understanding of language order and structure.
- BART (Lewis et al., 2020) (Bidirectional and Auto-Regressive Transformers) combines the best of both auto-encoding and auto-regressive approaches, excelling in text generation and comprehension tasks by reconstructing text that has been corrupted, making it highly suitable for complex comprehension and synthesis tasks.
- RoBERTa (Robustly Optimized BERT Approach) (Liu et al., 2019) iterates on BERT by modifying key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates. This results in improved performance across a range of benchmark tasks.
- T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020) adopts a unified approach, treating every NLP problem as a text-to-text task, simplifying the process of applying a single model to a variety of tasks, thus streamlining the training and inference process for NLP models.

However, the training phase is structured to leverage the computational prowess of PyTorch, utilizing DataLoader for batch processing, and optimizing model performance with AdamW. We track correctness across epochs to gauge improvement, employing a stopping criterion based on minimal gains in validation accuracy to prevent overfitting. The process begins by selecting the appropriate tokenizer and model architecture based on predefined criteria. Each model is trained over several epochs, with performance on the validation set carefully monitored to ensure improvement. We adopted a learning rate between 2e-5 and 3e-5, training across 4 epochs with batches of 16 to balance efficiency and accuracy. Early stopping was implemented to halt training if validation accuracy showed minimal improvement, preventing overfitting. This process ensured each model, from BERT to T5, was precisely tuned to our dataset's details, focusing on meaningful performance gains.

3.5 Model Fine Tuning

Model fine-tuning was an important aspect of our approach, tailored to make use of the full potential of pre-trained language models. By carefully adjusting learning rates, batch sizes, and epochs, we ensured that each model was optimally adapted to the specifics of the task. Our fine-tuning process also involved a careful selection of layers to unfreeze, enabling the models to learn task-specific details without overfitting.

3.6 Prediction On Validation Set

At the very first phase, we divided the training data into 90:10 manner to get a validation set for the prediction. We utilized a specific class to assess the accuracy of transformer models like XLNet, BART, BERT, RoBERTa, T5 on validation dataset. This class predicts the correct answers by tokenizing question-choice pairs and evaluating them through the model to select the most probable answer. The effectiveness of each model is quantified by comparing predicted answers against actual labels, providing a direct measure of performance. This approach allows us to take decision for the next step.

3.7 Prediction On Test Set

By doing the previous step on validation dataset, we have got the performance analysis of each model and it makes us to observe which model has done best in this set. We choose the best performing model to conduct a prediction on the given test set for the competition for both sentence and word puzzle data.

Model	Before Data Merging	After Data Merging
Bert	91%	**87%**
RoBERTa	90%	82%
XLNet	93%	**85%**
BART	88%	79%
T5	76%	70%

Table 1: Performance on validation dataset

4 Result

Our research explored how different transformer models performed when tasked with solving two types of puzzles: sentence and word puzzles. Initially, we observed encouraging results from all models on the sentence puzzles, which were more abundant in our dataset. This success highlighted the models' proficiency in contexts where narrative clues guide the solution process. However, when we combined sentence and word puzzles into a single dataset, we noticed a significant drop in accuracy across all models (as shown in Table 1). This decline suggests that the models, while effective at processing longer, context-rich sentences, struggled with the brevity and ambiguity typical of word puzzles. This challenge was particularly evident in our competitive analysis phase, where the BERT model achieved 77% accuracy, and a subsequent re-evaluation with XLNet showed an improvement to 80% accuracy on the test set.

The better performance on sentence puzzles can be attributed to the models' inherent strengths. Both BERT and XLNet are designed to excel in understanding and processing complex narrative contexts, benefiting from extensive pre-training across diverse text types. This foundation enables them to navigate the intricate language of sentence puzzles more adeptly. On the other hand, word puzzles often rely on subtle wordplay and linguistic nuances less represented in the models' training data, posing a greater challenge..

The disparity in performance between puzzle types underscores a crucial insight: transformer models, despite their advanced capabilities, exhibit varying degrees of adaptability to different linguistic tasks. The initial high accuracy rates with sentence puzzles showcase their potential, while the subsequent drop in performance upon introducing word puzzles highlights areas for improvement, particularly in enhancing the models' versatility and ability to generalize across diverse language tasks. Our findings indicate a clear path forward—further refining these models to better capture and interpret the breadth of human language, extending their applicability beyond structured narrative contexts to include the nuanced, often unpredictable areas of word puzzles.

5 Limitation and Error Analysis

Our study had some challenges and places where error analysis showed that things could be done better. One big problem with the models is that they are skewed because they were trained on datasets that might not fully show how people use words in different situations. This might make it harder for the models to generalise to new types of data, especially when they move from sentence puzzles to word puzzles (see Tables 2 and 3). The models' different results on sentence puzzles versus word puzzles also points out a need for error analysis and better training strategies or model architectures that can handle the complex nature of both types of puzzles equally. The fact that accuracy went down when datasets were combined suggests overfitting, an important problem that needs more research to make models more reliable. These new ideas help us plan future research, like looking into bigger and more varied training datasets and making models that are specifically made to deal with the problems that come up in different language tasks.

6 Conclusion

Our research into using transformer models like BERT, XLNet, and BART for question-answering tasks shows both their strengths and weaknesses when trying to understand words like humans do. The results point to a potential way to improve system's ability to interpret, but they also show that more progress needs to be made. In the future, action should be put into improving these models so that they understand context better, are more clear, and can be used in more areas. To close the gap between what we can do now and how well we can understand complex human language,

Phase	S_ori	S_sem	S_con	S_ori_sem	S_sem_con	S_overall
Competition	0.77	0.70	0.77	0.70	0.62	0.75
Post-Competition	0.80	0.75	0.77	0.75	0.65	0.77

Table 2: Performance metrics across for Sentence Puzzle

Phase	S_ori	S_sem	S_con	S_ori_sem	S_sem_con	S_overall
Competition	0.37	0.46	0.37	0.34	0.12	0.40
Post-Competition	0.56	0.53	0.40	0.50	0.25	0.50

Table 3: Performance metrics across for Word Puzzle

we will need to work together to improve model designs, training methods, and the way we combine different types of data. Not only does this project look like it will make NLP applications smarter, but it also opens up new ways for Computers to process and come up with language-based responses.

References

Saroj Bala. 2014. Lateral thinking vs vertical thinking. *Deliberative Research*, 24(1):25.

Tamer Boyacı, Caner Canyakmaz, and Francis de Véricourt. 2023. Human and machine: The impact of machine input on decision making under cognitive limitations. *Management Science*.

Edward De Bono. 1970. Lateral thinking. *New York*, page 70.

Phuc Do and Truong HV Phan. 2022. Developing a bert based triple classification model using knowledge graph embedding for question answering system. *Applied Intelligence*, 52(1):636–651.

Hadi Abdi Ghavidel, Amal Zouaq, and Michel C Desmarais. 2020. Using bert and xlnet for the automatic short answer grading task. In *CSEDU (1)*, pages 58–67.

Muhammad Jawad Hashim, Romona Govender, Nadirah Ghenimi, Alexander Kieu, and Moien AB Khan. 2023. Lectureplus: a learner-centered teaching method to promote deep learning. *Advances in Physiology Education*, 47(2):175–180.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Mika Koivisto and Simone Grassini. 2023. Best humans still outperform artificial intelligence in a creative divergent thinking task. *Scientific reports*, 13(1):13601.

Kamlesh Kumar, Prince Kumar, Dipankar Deb, Mihaela-Ligia Unguresan, and Vlad Muresan. 2023. Artificial

intelligence and machine learning based intervention in medical infrastructure: a review and future trends. In *Healthcare*, volume 11, page 207. MDPI.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020. Question and answer test-train overlap in open-domain question answering datasets. *arXiv preprint arXiv:2008.02637*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Khalid Nassiri and Moulay Akhloufi. 2023. Transformer models used for text-based question answering systems. *Applied Intelligence*, 53(9):10602–10635.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

FtG-CoT at SemEval-2024 Task 9: Solving Sentence Puzzles Using Fine-Tuned Language Models and Zero-Shot CoT Prompting

Micah Zhang **Shafiuddin Rehan Ahmed** **James H. Martin**
University of Colorado, Boulder, CO, USA
micah.zhang@colorado.edu

Abstract

Recent large language models (LLMs) can solve puzzles that require creativity and lateral thinking. To advance this front of research, we tackle SemEval-2024 Task 9: BRAINTEASER: A Novel Task Defying Common Sense. We approach this task by introducing a technique that we call Fine-tuned Generated Chain-of-Thought (FtG-CoT). It is a novel few-shot prompting method that combines a fine-tuned BERT classifier encoder with zero-shot chain-of-thought generation and a fine-tuned LLM. The fine-tuned BERT classifier provides a context-rich encoding of each example question and choice list. Zero-shot chain-of-thought generation leverages the benefits of chain-of-thought prompting without requiring manual creation of the reasoning chains. We fine-tune the LLM on the generated chains-of-thought and include a set of generated reasoning chains in the final few-shot LLM prompt to maximize the relevance and correctness of the final generated response. In this paper, we show that FtG-CoT outperforms the zero-shot prompting baseline presented in the task paper and is highly effective at solving challenging sentence puzzles achieving a perfect score on the practice set and a 0.9 score on the evaluation set.

1 Introduction

The BRAINTEASER SemEval-2024 Task (Jiang et al., 2023, 2024) explores the ability of large language models (LLMs) to perform lateral thinking or "thinking outside the box", a topic that is currently under-explored by the natural language processing (NLP) community. Unlike vertical thinking tasks that rely only on "common sense", solving this task requires a creative thinking process. The goal is to force LLMs to challenge their preconceptions and consider new perspectives. The task organizers propose a way to assess the ability of LLMs to think outside the box by creating a multiple-choice question-answering task designed to defy default commons-sense associations. For this task, the task organizers created a sentence puzzle dataset that contains sentence-type brain teasers centered on sentence snippets, and a word puzzle dataset that contains word-type brain teasers centered on the letter composition of the target question. Both datasets are written in English.

Our approach, Fine-tuned Generated Chain-of-Thought (FtG-CoT), as depicted in Figure 1, is a novel few-shot prompting method that combines a fine-tuned BERT classifier encoder with zero-shot chain-of-thought (CoT) generation and a fine-tuned GPT-3.5 LLM. The BERT (Devlin et al., 2019) classifier is fine-tuned by treating the training set as a multi-class classification task. Each training set question and choice list is treated as the classification example. The index corresponding to the correct answer choice is the class label. Reasoning chains for each example in the training set are generated using a variation of the classic zero-shot chain-of-thought prompt.

For each example, in addition to the question and choice list, the LLM is provided with the correct answer and asked to generate an explanation of why that is the correct answer. The generated chains of thought are then used to fine-tune the LLM. For each example in the fine-tuning dataset, the prompt contains the question and choice list, and the response contains the generated chain-of-thought. To get the final generated response, the fine-tuned LLM is queried with a few-shot prompt that contains a set of generated reasoning chains as example demonstrations.

The set of training demonstrations provided in the few-shot prompt is chosen based on their cosine similarity to the test question. Only the top 20 most similar training demonstrations are provided, ranked in order of increasing similarity. We chose to use OpenAI's GPT-3.5 Turbo model as the pre-trained LLM due to the ability to easily query and

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1245–1251

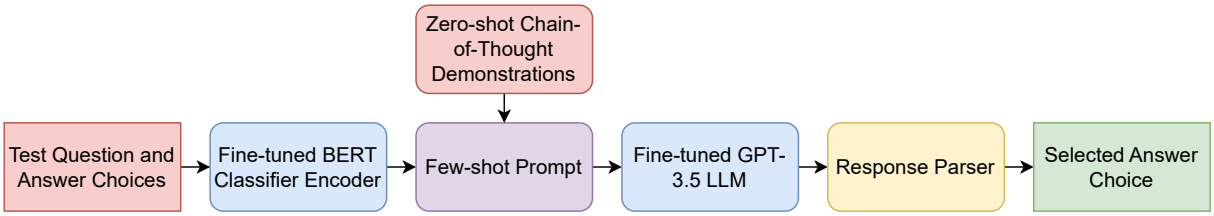


Figure 1: System diagram for FtG-CoT. To start, the test question and choice list are fed to a fine-tuned BERT classifier. The BERT classifier outputs an encoding that is used to identify the top 20 training questions that are most similar to the test question. The corresponding training demonstrations are combined in order of increasing similarity with the test question and choice list to create a few-shot prompt. The few-shot prompt is sent to a fine-tuned LLM, which outputs a generated response. A text parser matches the generated response with the most similar answer choice from the choice list.

fine-tune the model via the OpenAI API.

We provide a robust system that combines fine-tuned BERT and GPT-3.5 together with zero-shot CoT generation and ranked few-shot prompting. We found that this approach significantly improves the ability of the LLM to solve brain teaser sentence puzzles compared to the baseline zero-shot prompting approach, achieving a perfect score on the practice set and a 0.9 score on the evaluation set. Our method ranked 18th in the competition. We provide our code here: https://github.com/Micah-Zhang/SemEval-2024

2 Background

The BRAINTEASER task (Jiang et al., 2023, 2024) consists of two sub-tasks: Sentence Puzzles and Word Puzzles. Both datasets are written in English and each question is structured in a multiple-choice question format. Sentence Puzzles tend to be in a longer narrative story format. The correct answer challenges the common-sense interpretation of the question and the common-sense answer to the question. For Word Puzzles, the correct answer challenges the default meaning of a particular word and focuses on the letter composition of the question. Both types of questions provide 4 different answer choices, including 1 correct answer and 3 distractors. Figure 2 provides an example of a Sentence Puzzle and an example of a Word Puzzle for comparison.

We chose to participate in the Sentence Puzzle sub-task. For this sub-task, a training dataset consisting of 507 sentence puzzles was provided by the task organizers. The training set contained 169 original sentence puzzle examples obtained from public websites via web crawlers. Each example consisted of a question, a list of answer choices, the correct answer, and the distractors in the choice list. For each original question, a corresponding semantic reconstruction question and a context reconstruction question were generated. Semantic Reconstruction rephrases the original question without changing the correct answer and provided answer choices. Context Reconstruction changes both the question and answer to fit a new situational context, while keeping the original premise intact. The two adversarial question counterparts were created by the organizers to prevent an LLM from being able to win the competition using memorization only.

In addition to the training set, the organizers provide a practice evaluation and an official evaluation dataset for the Sentence Puzzle sub-task. Both datasets contain 120 questions. To evaluate a model, a text file containing the answer choice index for each selected answer choice is submitted to CodaLab. CodaLab then automatically calculates the corresponding accuracy scores and posts the results on the leaderboard.

Our approach is inspired by Wei et al. (2023), who introduced chain-of-thought prompting and demonstrated that providing a LLM with a series of intermediate reasoning steps improves its ability to perform complex reasoning. Our approach was also inspired by Kojima et al. (2023), who introduced zero-shot chain-of-thought prompting and demonstrated that LLMs could be made to generate their own series of intermediate reasoning steps by adding the words "Let's think step by step" to the end of the prompt. BERT (Devlin et al., 2019) is a popular transformer architecture commonly used as an sentence encoder for NLP tasks. The pre-trained BERT model can be fine-tuned by adding a classifier head to the end of the model and training it on a classification dataset. LLMs based on the GPT-3.5 architecture were popularized by Brown et al. (2020), who introduced few-shot prompting

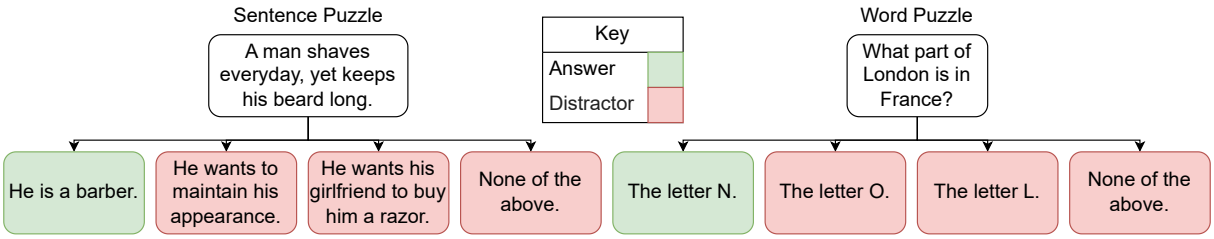


Figure 2: Diagram comparing Sentence Puzzles to Word Puzzles from the BRAINTEASER dataset. Sentence Puzzles tend to be in a longer narrative story format. The correct answer challenges the common-sense interpretation of both the question and its answer. For Word Puzzles, the correct answer challenges the default meaning of a word and focuses on the letter composition of the question.

and demonstrated that pre-trained LLMs could be made to solve new unseen tasks without needing additional training if the prompt included a set of example demonstrations.

3 System Overview

Fine-tuned Generated Chain-of-Thought Prompting (FtG-CoT) is a few-shot prompting method we developed for this competition. It consists of five steps. The first step is to fine-tune a pre-trained BERT-Small model on the training set as a multi-class classification task to use as an encoder. The second step is to use zero-shot chain-of-thought prompting (Kojima et al., 2023) to generate explanations for each demonstration in the training set. The third step is to fine-tune a large language model (LLM) on the generated chain-of-thought (Wei et al., 2023) demonstrations. The fourth step is to rank each encoded training demonstration based on its cosine similarity with the current encoded test question. The fifth step is to construct a few-shot (Brown et al., 2020) prompt by stacking the top N most similar demonstrations with the test question prompt in order of increasing similarity.

This few-shot prompt is sent to the LLM, an answer choice is extracted from the LLM's response, and the process is repeated for each question in the test set. This system is illustrated in Figure 1.

Fine-tuning BERT: We fine-tuned a pre-trained BERT model by converting it to a multi-class classifier and training it on the provided BRAINTEASER training set. This allowed us to create an encoder purpose-suited for solving sentence puzzles. During training the BERT classifier learns contextual and semantic information that is relevant to solving the sentence puzzle task. When used to encode a training or test example, this information is captured in the encoding and aids in selecting highly relevant and useful demonstrations to use for few-shot prompting.

Zero-shot prompting: Zero-shot prompting is used to generate chain-of-thought reasoning explanations for each training example. This is accomplished by prompting the LLM using a custom zero-shot prompt for each training example. The first three lines of the prompt contain the training question, answer choices, and correct answers. The fourth line is "Let's think step by step. What is the best answer? Explain in detail why this is the best answer in 5 or less sentences".

A custom fuzzy logic answer extractor parses the generated response from the LLM by isolating the sentence containing the selected answer and then using a sequence matcher to compare the generated answer against each of the answer choices in the choice list. The index of the answer choice that most closely matches the generated answer is set as the predicted label.

Fine-tuning GPT-3.5: Fine-tuning a pre-trained GPT-3.5 model on the 507 training examples and their corresponding generated reasoning chains improves the accuracy and consistency of the answers generated by the LLM. The fine-tuning dataset contains an example prompt and response for each training question. The first two lines of the prompt contain the question and answer choices. The third line is "Let's think step by step. What is the best answer?". The ground truth response is the generated response from the previous zero-shot prompting step.

Ranking Demonstrations: Cosine similarity is defined as $cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$, where A represents the vector encoding of a test question and B represents the vector encoding of a training question, both produced using the fine-tuned BERT classifier. Providing the top N training examples that have the highest cosine similarity with test questions as few-shot demonstrations ensures that the selected

demonstrations are highly relevant to the current test question.

Few-shot Prompt: The final prompt sent to the fine-tuned LLM contains all N demonstrations ranked in increasing order of similarity, the test question and answer choices, as well as the line "Let's think step by step. What is the best answer?". The received response is parsed using the same fuzzy logic sequence-matching answer extractor.

4 Experimental Setup

Training FtG-CoT requires fine-tuning BERT and fine-tuning GPT-3.5. We use all 507 examples from the sentence puzzle training set provided for the BRAINTEASER task for fine-tuning.

To fine-tune BERT, we start with the BERT-Small pre-trained model from Google (Bhargava et al., 2021) and add a multi-class classifier head consisting of two linear hidden layers separated by a ReLU activation function to create a BERT classifier. The input to the classifier head is the [CLS] token from BERT: the first token from BERT's final hidden layer. The first hidden layer in the classifier head projects the BERT output from dimensions 512 to 256. The output of the second hidden layer is passed through a log softmax function.

To train the BERT classifier on the Sentence Puzzle training set, we first concatenate the question and choice list from each example and provide them as the input string to the classifier. The index of the correct answer from the choice list is the label for the classifier. During training the BERT classifier encodes each input string via the pre-trained BERT model and passes the encoding to the classifier head, which then assigns a class label to the input. The model calculates the negative log-likelihood loss between the predicted and true label and back-propagates the loss through the entire model, including the BERT layers. The BERT classifier was trained using an AdamW optimizer for 40 epochs with a learning rate of 0.00001. The final layer of the trained BERT classifier produces encodings that have been fine-tuned to solve sentence puzzles.

To fine-tune GPT-3.5, we start with the gpt-3.5-turbo-1106 pre-trained model from OpenAI and fine-tune the model using the OpenAI fine-tuning API. We create a fine-tuning dataset according to the format required by the OpenAI API. For each of the 507 training examples the system is set to "teacher", and the user prompt is defined as a concatenation of the question and choice list. The ground truth assistant response is defined as the generated chain of thought. The fine-tuned GPT-3.5 model was trained for 3 epochs with a batch size of 1 and a learning rate multiplier of 2. All other training hyperparameters cannot be set externally by the user and are instead defined internally by OpenAI.

The model is evaluated using the test set provided for the BRAINTEASER task. The test set consist of 120 questions, each with their own choice list. For each of the 120 questions, we use FtG-CoT to create a few-shot prompt used to query the fine-tuned GPT-3.5 model via the OpenAI API. We set the temperature to 1.0. The corresponding received response is then parsed using the custom fuzzy logic answer extractor described earlier. The predicted labels for all 120 test questions are then submitted to the BRAINTEASER CodaLab website where it is automatically graded against the ground truth labels. The resulting accuracy score is displayed on the competition leaderboard.

5 Results

The task organizers created 6 different accuracy metrics for evaluation. The first 3 metrics are instance-based accuracy scores that measure the accuracy of the model in solving the original questions, the semantic reconstruction questions, and the context reconstruction questions separately from one another. The next 2 metrics are group-based accuracy scores that consider each original puzzle and its variants as a single group. For each group, the model will score 1 accuracy point only if it successfully solves all three puzzles in the group. Otherwise, it will score 0 points. The last metric is an overall accuracy that is calculated as the average of the 3 instance-based accuracy scores.

FtG-CoT performs well at the task according to the official metrics, achieving a perfect score on the practice set and a 0.9 score on the evaluation set, ranking 18th overall in the competition. In this context, "score" refers to the accuracy score for the original practice and evaluation questions, not including the semantic and context reconstruction questions.

Table 1 compares the official leader board results for FtG-CoT evaluated on sentence puzzle evaluation dataset against the zero-shot ChatGPT baseline provided by the BRAINTEASER task organizers. FtG-CoT outperformed the zero-shot baseline for

Method	Original	Semantic	Contextual	Ori + Sem	Ori + Sem + Con	Overall
Zero-shot	0.608	0.593	0.679	0.507	0.397	0.627
FtG-CoT	0.900	0.825	0.775	0.800	0.675	0.833

Table 1: Official leader board results for sentence puzzle evaluation dataset. Compares FtG-CoT performance against the zero-shot ChatGPT baseline provided by the BRAINTEASER task organizers. The six categories correspond to the three instance-based accuracy scores, the two group-based accuracy scores, and the overall accuracy score. FtG-CoT outperforms the baseline in each of these metrics.

all 6 metrics. It performed best at answering the original question and struggled the most with the combined semantic and original group-based accuracy metric.

Method	Score (Original)
1-shot	0.8
5-shot	0.925
10-shot	0.925
10-shot w/ reversed order	0.90
20-shot	0.925
10-shot w/ fine-tuning	0.975
20-shot w/ fine-tuning	1.0
30-shot w/ fine-tuning	0.975

Table 2: Experimental results comparing the performance of FtG-CoT on the sentence puzzle practice dataset with different configurations. In general, increasing the number of demonstrations, listing them in order of increasing similarity, and fine-tuning the LLM on the training demonstrations tended to result in higher accuracy scores.

Table 2 displays practice test set scores achieved by FtG-CoT using different numbers of demonstrations, ordering, and with and without fine-tuning GPT-3.5 on the training set. By default, all demonstrations were concatenated in order of increasing similarity with the test question. In general, increasing the number of few-shot training demonstrations provided in the prompt tends to improve the performance of the LLM. However, past a certain number, in our case 10 demonstrations, adding additional demonstrations does not improve the performance of the LLM and results in decreased performance.

Furthermore, reversing the order of the demonstrations to be in order of decreasing similarity such as that the least similar demonstration is located closest in proximity to the test question in the prompt resulted in a lower score. This suggests that listing few-shot demonstrations in order of increasing similarity is the better approach.

The results also illustrate the effectiveness of fine-tuning the LLM on the training set. Whereas without fine-tuning the score peaked at around 0.925 regardless of the number of demonstrations provided, fine-tuning the LLM immediately resulted in a jump in performance to 0.975. Fine-tuning the LLM combined with increasing the number of provided demonstrations to 20 resulted in the highest achieved score of 1.0. However, increasing the number of provided demonstrations past 20 did not result in additional improved performance.

6 Error Analysis

For error analysis, we performed a 80/20 randomized train/test split on the sentence puzzle training set to create our own error analysis test set. We then fine-tuned FtG-CoT on the remaining training examples and evaluated the model against this test set.

Early analysis revealed that many of the questions that the model answered incorrectly were false negatives resulting from using exact string matching for answer extraction. This motivated the creation of the fuzzy logic answer extractor. We also found that the LLM struggled with answer consistency and repeatability. Fine-tuning the model helped to lessen this issue. However, even with the fuzzy extractor and fine-tuning, the LLM was not able to achieve perfect accuracy on the error analysis test set.

Figure 3 provides both an example of a sentence puzzle that the model answered correctly as well as an example of a sentence puzzle that the model answered incorrectly. Both questions belong to the same group of sentence puzzles. The question the model answered correctly was the original sentence puzzle. The question the model answered incorrectly was the semantically reconstructed version of the sentence puzzle. The semantically reconstructed question rephrases the original question, keeps the same correct answer, and shuffles the provided answer choices.

In this case, both questions share the same premise: a group of animals are racing to grab

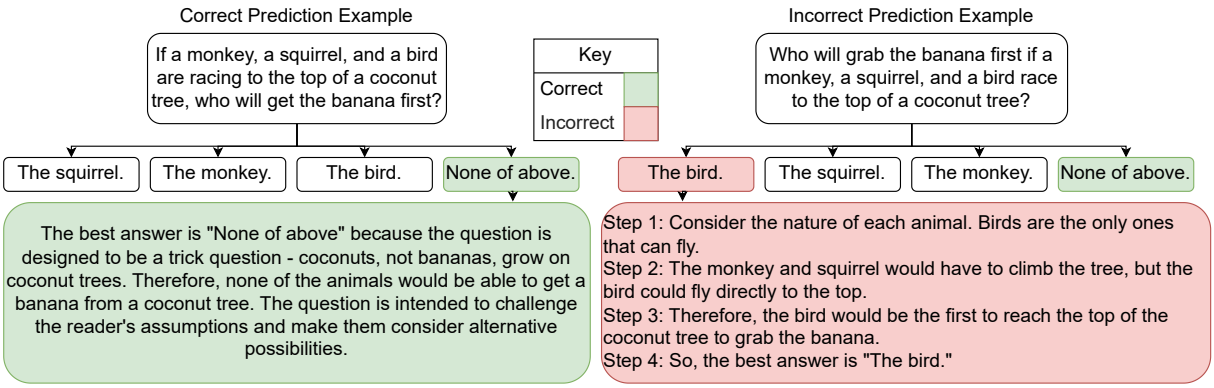


Figure 3: Example of a generated response with the correct answer for an original sentence puzzle as well as an example of a generated response with the incorrect answer for its corresponding semantically reconstructed sentence puzzle. The LLM correctly identifies the puzzle twist behind the original sentence puzzle and uses it to arrive at the correct answer, but is unable to do so for the semantically reconstructed sentence puzzle.

a fruit at the top of a tree, which animal will reach the fruit first? They also share the same puzzle twist: we are told that the animals are climbing a coconut tree yet are asked which animal will reach the banana first. Since bananas cannot grow on coconut trees, none of animals will be able to reach the banana. Therefore, the correct answer is "None of the above".

The generated response for the original sentence puzzle demonstrates that the model was able to correctly identify the puzzle twist and use it to arrive at the correct answer. However, the generated response for the semantically reconstructed question demonstrates that the model was not able to apply this same reasoning to the rephrased question. Instead, it answers the question as if it was asking about a coconut and a coconut tree and bases its answer off of the speed and movement of the animals, choosing the bird for its ability to fly.

These examples illustrate how the fine-tuned LLM model is highly sensitive to word choice and order, which helps explain why the model performs best on the original sentence puzzles and tends to struggle with the semantically and contextually reconstructed versions of the original puzzles. It is possible that the dataset OpenAI used to train GPT-3.5 contains a portion of the original sentence puzzles in the error analysis test set, allowing it to rely on memorization to improve its performance. However, since the LLM's accuracy on the reconstructed questions is only around 10% lower compared to the original questions, this suggests that the model is not relying solely on memorization to solve the sentence puzzles and may be capable of lateral thinking.

7 Conclusion

The FtG-CoT few-shot prompting method we developed for this competition combines a fine-tuned BERT classifier encoder with zero-shot chain-of-thought generation and a fine-tuned LLM. Our experiments have demonstrated that FtG-CoT is highly effective at solving sentence puzzles, achieving a perfect score on the practice set and a 0.9 score on the evaluation set, ranking 18th overall in the competition.

The key takeaways from our experiments are that FtG-CoT performs better with a larger number of demonstrations up to a certain point, few-shot demonstrations should e listed in order of increasing similarity, fine-tuning results in markedly improved performance, and that FtG-CoT significantly outperforms zero-shot prompting on the sentence puzzle evaluation set.

Regarding future work, it is possible that the performance of FtG-CoT can be further improved by using data augmentation techniques to expand the training set. For example, prompting an LLM to rephrase the existing training questions could increase the size of the training set and potentially decrease the fine-tuned model's sensitivity to word choice and order. It is also possible that the performance of FtG-CoT can be further improved by incorporating human feedback via reinforcement learning. A human could provide feedback on the quality and accuracy of the generated reasoning chains as well as manually rewrite incorrectly generated reasoning chains, improving the quality of both the fine-tuning dataset and the demonstrations.

8 Acknowledgements

We would like to thank Jie Cao and Abteen Ebrahimi for their guidance and feedback. We would also like to extend our thanks to Yifan Jiang, Filip Ilievski, and Kaixin Ma for creating the BRAINTEASER SemEval-2024 Task.

References

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

LyS at SemEval-2024 Task 3: An Early Prototype for End-to-End Multimodal Emotion Linking as Graph-Based Parsing

Ana Ezquerro and David Vilares
Universidade da Coruña, CITIC
Departamento de Ciencias de la Computación y Tecnologías de la Información
Campus de Elviña s/n, 15071
A Coruña, Spain
{ana.ezquerro, david.vilares}@udc.es

Abstract

This paper describes our participation in SemEval 2024 Task 3, which focused on Multimodal Emotion Cause Analysis in Conversations. We developed an early prototype for an end-to-end system that uses graph-based methods from dependency parsing to identify causal emotion relations in multi-party conversations. Our model comprises a neural transformer-based encoder for contextualizing multimodal conversation data and a graph-based decoder for generating the adjacency matrix scores of the causal graph. We ranked 7th out of 15 valid and official submissions for Subtask 1, using textual inputs only. We also discuss our participation in Subtask 2 during post-evaluation using multi-modal inputs.

1 Introduction

SemEval 2024 Task 3 focused on Multimodal Emotion Cause Analysis in Conversations (Wang et al., 2024). Figure 1 shows an example provided by the organizers to illustrate the task. Two subtasks were proposed: Subtask 1, which uses only textual inputs, and Subtask 2, which allows for the consideration of video and audio processing as well.

The shared task is timely given the recent success of multimodal architectures combining computer vision (Redmon et al., 2016; Wang et al., 2023b), natural language processing (Devlin et al., 2019; Beltagy et al., 2020), and speech processing (Gong et al., 2021; Radford et al., 2022) advancements. In the particular context of multimodal emotion analysis, the task builds on top of previous work such as recognizing the triggered emotions as a classification task (Alhuzali and Ananiadou, 2021; Zheng et al., 2023) or predicting complex cause-effect relations between speakers (Wei et al., 2020; Ding et al., 2020). For the particular case of the shared task, the dataset - centered in English - relies on (Wang et al., 2023a) and provides text, image and audio inputs.

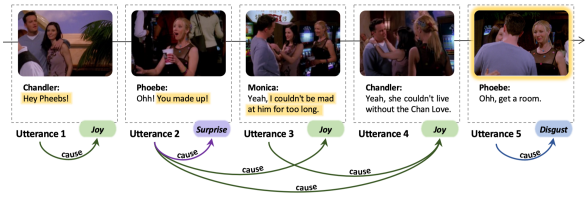


Figure 1: Example taken from the official website of the SemEval Task 3 - https://nustm.github.io/SemEval-2024_ECAC/. The goal of the task consists of predicting (i) the emotion associated to each utterance within the conversation, (ii) the cause-effect relations that trigger the emotions between utterances and (iii) the associated span in the cause utterance.

We had time and resources only to build a textual model for official participation in Subtask 1. We validated some multimodal baseline approaches using vision and audio inputs, but the computational resources required to fine-tune text and video data were beyond our range, so we participated in Subtask 2 only during post-evaluation. In what follows, we describe our approach. The implementation of our early prototype can be found at https://github.com/anaezquerro/semeval24-task3.

Contribution We propose an end-to-end multimodal prototype based on a large multimodal encoder to contextualize text, image and audio inputs with a graph-based decoder to model the cause-effect relations between triggered emotions within multi-party conversations. The large encoder joins pretrained architectures in text (Devlin et al., 2019), image (Dosovitskiy et al., 2021) and audio (Baevski et al., 2020) modalities, while the decoder is adapted from the graph-based approaches in semantic parsing (Dozat and Manning, 2018). The model is trained end-to-end.

2 Background

Multimodal Emotion Cause Analysis A number of datasets collecting multi-party conversations

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1252–1259

(Poria et al., 2019; Chatterjee et al., 2019; Firdaus et al., 2020) have been published to train and test multimodal neural architectures. Simpler configurations involve recognizing the speaker emotion at each utterance - this task is commonly known as Emotion Recognition (ER) (Poria et al., 2019) - while others require a deeper level of understanding to model interactions and causal relations between speakers - Emotion-Cause Pair Extraction (ECPE) (Xia and Ding, 2019). The most common approaches follow an encoder-decoder neural architecture where the encoder is conformed by multiple modules - one module per input modality (text, image and/or audio) - and produces an inner representation at utterance level; and the decoder accepts the encoder outputs as inputs and returns a suitable output adapted to the specifications of the targeted task. In the context of Multimodal ER, Nguyen et al. (2023) proposed a GCN-based decoder to capture temporal relations (Schlichtkrull et al., 2017), while Dutta and Ganapathy (2024) used cross-attention to fusion the input modalities and a final classification layer to predict the targeted emotions. Approaches in ECPE require an extra effort to represent and model causal information: Wei et al. (2020) scored all possible utterance tuples to predict the most probable list of emotion-cause pairs. Other authors, like Chen et al. (2020) and Fan et al. (2020), represented the emotion-cause pairs as a labeled graph between utterances and tried to predict the set of causal edges using a GCN or a transition-based system, respectively. The SemEval 2024 Task 3 joins the recognition and causal extraction tasks and challenges a system able to both model speaker emotions and elicit relations.

Graph-based decoding For structured prediction tasks, such as dependency parsing, graph-based approaches are a standard for computing output syntactic representations (McDonald, 2006; Martins et al., 2013). Particularly, Dozat and Manning (2017) introduced a classifier that computes a head and dependent representation for each token and then uses two biaffine classifiers: one computes a score for each pair of tokens to determine the most likely head, and the other determines the label for each head-dependent token pair. We will also build upon a biaffine graph-based parser: we will frame the task as predicting a dependency graph, where utterances are the nodes and emotions are dependency labels between pairs of utterances.

3 System Overview

Our system consists of two modules: a large pre-trained encoder and a graph-based decoder (see Figure 2). It can add extra input channels into the encoder without requiring any adjustments to the decoder, so the same decoder is used for both tasks, while the encoder is adapted to incorporate text-only (Subtask 1) or multimodal (Subtask 2) inputs.

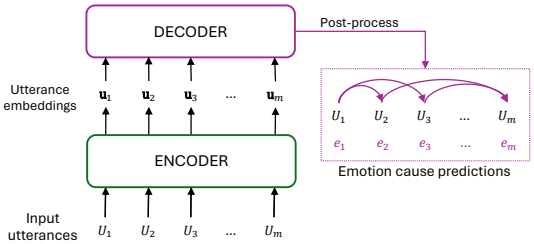


Figure 2: High-level architecture of our system. The encoder takes as input the sequence of m utterances of a given conversation and returns a unique vector representation for each utterance. The decoder uses the utterance embedding matrix to apply the affine attention product in the decoder, obtain the scores of the adjacent matrix and return the predicted sequence of emotions and the cause relations between utterances.

Let $C = (U_1, ..., U_m)$ be a conversation of m utterances, where each utterance $U_i = \{W_i, s_i, \varepsilon_i\}$ is defined by (i) a sequence of words $W_i = (w_1^{(i)}, ..., w_{\ell|w,i|}^{(i)})$[1], (ii) an active speaker s_i and (iii) a triggered emotion $\varepsilon_i \in \mathcal{E}$[2]. The set of cause-pair relations between utterances can be represented as a directed labeled graph $G = (\mathcal{U}, \mathcal{R})$ where $\mathcal{U} = (U_1, ..., U_m)$ is the sequence of utterances of the conversation assuming the role of the nodes of the graph and $\mathcal{R} = \{U_i \xrightarrow{\varepsilon_j} U_j,\ i, j \in [1, m]\}$ is the set of emotion-cause relations between an arbitrary cause utterance U_i and its corresponding effect U_j. Thus, the task can be cast as the estimation of the adjacent matrix of G, similarly to syntactic (Ji et al., 2019) and semantic dependency (Dozat and Manning, 2018) parsing. Adapting algorithms from parsing to model emotion-cause relations between utterances has also been explored by other authors, such as Fan et al. (2020), who instead explored a transition-based strategy.

[1]From now on, we denote as $\ell|\cdot, i|$ the length of the i-th in a sequence $\cdot$, so $\ell|w, i|$ denotes the length of the W_i. Table 2 summarizes the notation used in this paper.

[2]The set of emotions are described in Wang et al. (2023a).

3.1 Textual Extraction

The first subtask draws from only textual information to predict the adjacent matrix of G with a span that covers the specific words from U_i that trigger the emotion ε_j in the cause relation $U_i \xrightarrow{\varepsilon_j} U_j$.

Textual encoder Figure 3 illustrates our encoder. Given the sequence of utterances $(U_1, ..., U_m)$, we encoded with BERT the batched sequence of utterances where each word sequence was preceded by the CLS token (Devlin et al., 2019). For each U_i, we select the CLS embedding $(\mathbf{u}_i)$ from the contextualized embedding matrix $\mathbf{W}_i = (\mathbf{u}_i, \mathbf{w}_1, ..., \mathbf{w}_{\ell|w,i|})$, which is assumed to have information of the whole sentence. The CLS embedding matrix $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_m)$ was passed as input to the decoder module and the word embeddings were reserved for the span attention module.

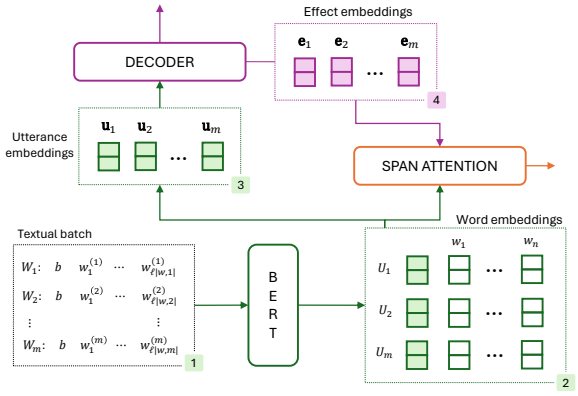


Figure 3: High level representation of the textual encoder. The input (1) is the matrix of stacked token vectors of each utterance. The last hidden states of BERT are used as word embeddings (2) and the special CLS tokens are used as utterance embeddings (3). The effect embeddings (4) - a partial representation from the decoder - are taken as input to the span module with the contextualized BERT embeddings.

Graph-based decoder Figure 4 shows the forward-pass of the graph-based decoder from the encoder output of Figure 3. To produce an adjacent matrix $\mathbf{G}$ of dimensions $m \times m$, where each position (i, j) represents the probability of a causal relation from U_i (cause) to U_j (effect), the first biaffine module uses a trainable matrix $\boldsymbol{\mathcal{W}}_{\mathrm{G}} \in \mathbb{R}^{d_{\mathrm{G}} \times d_{\mathrm{G}}}$ and maps $\mathbf{U}$ using two feed-forward networks to a cause $(\mathbf{C})$ and an effect $(\mathbf{E})$ representation. By projecting the original BERT embeddings to two different representations, $\mathbf{u}_i \sim (\mathbf{c}_i, \mathbf{e}_i)$, the decoder learns different contributions for the same utterance depending on the role. The affine product is

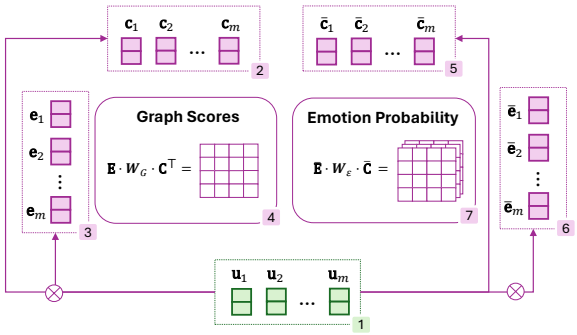


Figure 4: Graph-based decoder. The utterance embeddings (1) are projected to different representations (2 , 3 , 5 6) using four feed-forward networks to flexibly represent utterance embeddings. The scores of the adjacent matrix and the probability tensor are computed with the affine attention product.

defined as $\mathbf{G} = \mathbf{E} \cdot \boldsymbol{\mathcal{W}} \cdot \mathbf{C}^{\top}$. The second biaffine module uses a trainable tensor $\boldsymbol{\mathcal{W}}_{\varepsilon} \in \mathbb{R}^{d_{\mathrm{G}} \times |\mathcal{E}| \times d_{\mathrm{G}}}$ to predict the probabilities of triggered emotions between cause-effect utterances.

Span Attention module To maintain the end-to-end prediction while learning the span associated to each relation $U_i \rightarrow U_j$, we created a binary tensor $\mathbf{S} = (\mathbf{S}_1 \cdots \mathbf{S}_m)$ of dimensions $m \times m \times \max_{i=1,...,m}\{\ell|w,i|\}$[3] to specify if a word $w_k \in W_i$ of U_i is included in the span that triggers an emotion in U_j. To compute each $\mathbf{S}_i$, the matrix of word embeddings $(\mathbf{W}_i)$ of the utterance U_i is passed through a One-Head Attention module (see Figure 5), where $\mathbf{W}_i$ acts as the query matrix and $\mathbf{E}$ as the key and value matrices, so $\mathbf{S}_i = \Phi(\mathrm{softmax}(\mathbf{W}_i \cdot \mathbf{E}^{\top}) \cdot \mathbf{E})$, where Φ is a feed-forward network to project the embedding dimension to a unique binary value.

Encoding speaker information The dataset includes information about the active speakers in each utterance. A first approach to use this information as input would be concatenating the speaker embeddings to the sequence of utterances. However, this might lead to some issues: the model could assume that there is some inner dependency between triggered emotions and the characters in the conversation. This might be true in some cases, but it can also lead to biases, and there is still the challenge of modeling infrequent and unknown characters. To deal with this, we encoded a conversation C with speakers $s_1, ..., s_m$ using relative

[3]Note that each matrix $\mathbf{S}_i$ has dimensions $m \times \max_{i=1,...,m}\{\ell|w,i|\}$ and is associated to a *cause* utterance.

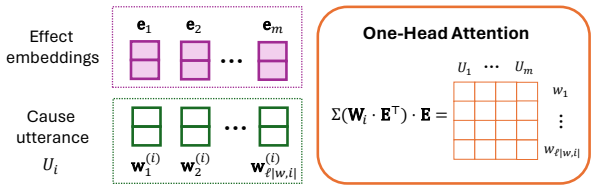


Figure 5: Span Attention module adapted from Vaswani et al. (2017). The tensor of word embeddings ($\mathbf{W}_1 \cdots \mathbf{W}_m$) from the encoder (Figure 3) and the effect contextualizations ($\mathbf{E}$) from the decoder (Figure 4) are passed to the attention product using each $\mathbf{W}_i$ as *key* and $\mathbf{E}$ as *query* and *value* matrices.

positional embeddings. For instance, the sequence (Chandler, Phoebe, Monica, Chandler, Phoebe) in Figure 1 would be encoded as $(0, 1, 2, 0, 1)$.

3.2 Multimodal Extraction

The second subtask adds a short video representation to each utterance, so U_i in a conversation $C = (U_1, ..., U_m)$ is now a tuple of five different elements $U_i = \{W_i, s_i, \varepsilon_i, \mathbf{X}_i, \mathbf{a}_i\}$. The last two added items encode the image and audio: (i) $\mathbf{X}_i = (\mathbf{x}_1^{(i)}, ..., \mathbf{x}_{\ell|x,i|}^{(i)})$ is the sequence of frames of the input video, where each frame is an image[4] tensor of dimensions $h \times w \times 3$ and (ii) $\mathbf{a}_i$ is the sampled audio signal of arbitrary length.

Image encoding We relied on a Transformer-based architecture (Ma et al., 2022; Zheng et al., 2023) to contextualize input images. While recent studies have proposed adaptations of the Vision Transformer and 3-dimensional convolutions that capture temporal correlations between sequences of frames for video classification (Arnab et al., 2021; Ni et al., 2022), our experiments were constrained by our resource limitations, preventing us from using these pretrained architectures. Hence, for our multimodal baseline we opted for the the smallest version of the Vision Transformer (ViT) model (Dosovitskiy et al., 2021) pretrained on the Facial Emotion Recognition dataset (Goodfellow et al., 2013)[5] to contextualize a small fraction of sampled frames[6], and incorporated an LSTM-based module to derive a unique image representation for each utterance. From an image batch $\mathbf{X}_i$, each image $\mathbf{x}_k^{(i)} \in \mathbb{R}^{h \times w \times 3}$ was passed to the ViT base model to recover the output of the last hidden layer and introduce it as input to the LSTM module to recover a final representation for U_i.

Audio encoding For our multimodal system we used the hidden contextualizations of the base version of wav2vec 2.0 (Baevski et al., 2020)[7]. Given a raw audio ($\mathbf{a}_i$) of an utterance U_i, the encoder of wav2vec 2.0 returns a sequence of hidden states that we summarized with an additional trainable LSTM layer to retrieve a unique vector that contents the audio information.

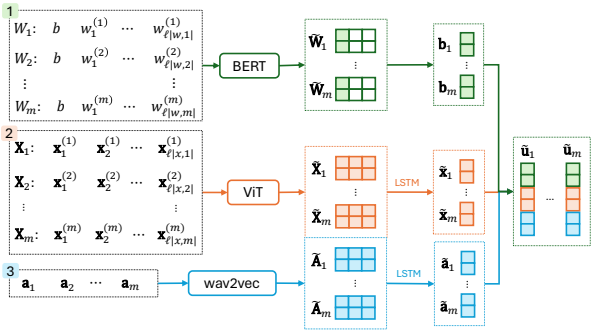


Figure 6: Multimodal encoder for Subtask 2.

Model fine-tuning The multimodal encoder (§3.2) uses three pretrained architectures to contextualize individual utterances and passes to the decoder the concatenation of the three unimodal representations (Figure 6). We chose to fine-tune only BERT during training together with the rest of the network. This was based on our empirical observation of superior results when learning from text compared to image and audio data. We entrusted the learning of audiovisual data to the LSTM learnable module within the encoder, presuming an accurate initial contextualization from wav2vec 2.0 and ViT pretrained on FER-2013.

3.3 Post-processing

Our end-to-end system directly recovers the predicted emotion-cause relations in a single post-processing step that linearly operates with the output tensors of the decoder. For the first subtask, the decoder returns (i) the adjacent matrix $\mathbf{G} \in \mathbb{R}^{m \times m}$, (ii) the labeled adjacent matrix $\overline{\mathbf{G}} \in \mathbb{R}^{m \times m \times |\mathcal{E}|}$ and (iii) the span scores $\mathbf{S} \in \mathbb{R}^{m \times m \times \ell_{\max}|w,i|}$. As Dozat and Manning (2017), each arc $U_i \rightarrow U_j$ is predicted by thresholding $\mathbf{G}$, and, once the arcs are predicted, the tensor $\overline{\mathbf{G}}$ determines the label

[4] All frames are RGB images, being the majority resolution 720×1280.

[5] https://huggingface.co/trpakov/vit-face-expression.

[6] For our experiments we used 5 interleaved frames per video, although a lower sampling rate can be considered depending on the computational capabilities.

[7] https://huggingface.co/facebook/wav2vec2-base-960h.

(emotion) associated to each arc. Since our formalization (§3.1) associates a given utterance to an unique emotion, we leveraged the scores of $\overline{\mathbf{G}}$ by the cause utterances and return the emotion with highest score. Finally, to produce a continuous span for each score vector $\mathbf{s_{ij}}$, we considered the leftmost and rightmost elements of $\mathbf{s}_{ij}$ higher than a fixed threshold.

ST-1	$\mathbf{P}^{\mathbf{w}}_{\mathbf{s}}$	$\mathbf{R}^{\mathbf{w}}_{\mathbf{s}}$	$^{\dagger}\mathbf{F}^{\mathbf{w}}_{\mathbf{s}}$	$\mathbf{P}^{\mathbf{w}}_{\mathbf{p}}$	$\mathbf{R}^{\mathbf{w}}_{\mathbf{p}}$	$\mathbf{F}^{\mathbf{w}}_{\mathbf{p}}$
BERT$_{400}$	10.19	5.46	7.01	21.64	15.09	17.33
BERT$_{600}$	12.61	7.43	9.32	22.06	15.2	17.95
BERT$_{800}$	14.89	7.36	9.75	22.13	23.25	15.32

ST-2	$\mathbf{P}^{\mathbf{w}}$	$\mathbf{R}^{\mathbf{w}}$	$^{\dagger}\mathbf{F}^{\mathbf{w}}$
BERT	27.49,	17.62,	20.43
+ViT	22.38	22.72	22.17
+w2v	28.4	20.01	23.36
+w2v+ViT	23.37	7.62	11.49

Table 1: Evaluation of our prototype with different multimodal configurations. Precision (P), recall (R) and F-Score (F) measured the weighted average across the eight emotions of the dataset (superscript **w** denotes that the measure is weighted) and for the first subtask the span performance is considered with strict correctness (subscript **s**) or overlapping (subscript **p**). The symbol † remarks the reference metric for each subtask.

4 Experiments

Validation The annotated dataset contains 1 375 multi-party conversations with a total of 13 619 utterances (Wang et al., 2023a). Although an unbiased estimation of the performance of our system would require validating the trained architecture using all available annotated data, our time and resources limitations prevented us from conducting k-fold cross-validation. Instead, we partitioned a 15% of the annotated dataset as our development set. The specific split used will be available with the accompanying code to replicate our findings.

Evaluation We use the official metrics[8]: the weighted strict F-Score for the Subtask 1 and the weighted F-Score for the Subtask 2.

Hyperparameter configuration Our computational limitations prevented us from exhaustively searching the optimal hyperparameters for our system. We conducted some tests varying the pre-trained text encoder[9], model dimension, gradient descent algorithm and learning rate and adding or removing the speaker module. We maintained in all experiments some regularization techniques (such as dropout in the hidden layers and gradient norm clipping) to avoid over-fitting. Our final configuration uses AdamW optimizer (Loshchilov and Hutter, 2019) with learning rate of 10^{-6} and is trained during one hundred epochs with early stopping on the validation set.

5 Results

Table 1 presents the performance of our system for both subtasks. For the first subtask, we investigated various embedding sizes of the Biaffine decoder while concurrently fine-tuning the largest version of BERT[10]. For the second subtask, we conducted experiments using different types of inputs to evaluate their impact. These included: (i) using only text-based inputs, (ii) adding audio data, (iii) incorporating visual data through frames, and (iv) leveraging all available multimodal inputs together. For approaches (i), (ii) and (iii), only BERT was fine-tuned, whereas for approach (iv), all pretrained weights were frozen. These weights solely served to contextualize input information, with the learning process confined to the decoder component.

Our top-performing model for the first subtask achieved a validation score of 9.75 and ranked in the evaluation set in 7th position among 15 participants with 6.77 points. We observed a slight performance improvement by increasing the hidden dimension of the decoder. Thus, considering the expansion of decoder layers could improve the performance. It is worth noting the significant impact of span prediction on the model performance: the proportional results consistently outperform strict metrics. Removing span prediction while retaining only text inputs results in a notable increase in F-Score (20.43 points for the second subtask), indicating the crucial role of span prediction in model learning. Furthermore, we noticed that there was a consistent delay in the alignment between recall and precision metrics, with precision consistently exceeding recall by more than 5 points across all approaches. This suggests that our system tends to adopt a conservative behavior, avoiding the number

[8] https://github.com/NUSTM/SemEval-2024_ECAC/tree/main/CodaLab/evaluation.

[9] We performed some experiments using all the versions of BERT, (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) and selected the best-performing textual encoder (BERT-large).

[10] https://huggingface.co/google-bert/bert-large-cased

of false cause emotion predictions.

The best validation performance for the second subtask is achieved through the integration of text and audio, yielding a score of 23.36 points in the weighted F-Score. Using image data also improves the text-only baseline, though unexpectedly lags behind the audio model. It is important to note that these two approaches are not directly comparable due to differences in their data inputs: the text and image model only considers a fixed number of sampled frames, suggesting that providing more image data (ideally, the full sequence of frames) could potentially yield a better performance that surpasses the audio-based approach. Unfortunately, we could not fine-tune BERT with the full multimodal encoder, so we were restricted to projecting the multimodal inputs to their respective contextualizations, and relying on the trainable weights of the decoder to optimize the full architecture. The results prove the importance of, at least, fine-tuning the text encoder: the F-Score only reaches 11.25 points, whereas the text finetuned baseline nearly doubles its performance with 20.43 points, highlighting the insufficient context of the original pretrained BERT embeddings to address this task.

Once the post-evaluation period concluded, we upload an experimental submission of our best multimodal system to the official competition. We obtained 15.32 points in the weighted F-Score, positioning our baseline in the 13th place out of 18 participants.

6 Conclusion

We proposed a graph-based prototype for the analysis emotion-cause analysis in conversations. Given the limited preparation time, we only submitted official results for Subtask 1 (text-only), but also report post-evaluation results for Subtask 2 (multimodal). The task required predicting several aspects of the conversation: (i) the emotion associated with each utterance, (ii) the cause-effect relationships triggering these emotions between utterances, and (iii) the specific span within the cause utterance responsible for the emotion. We achieved 7th place out of 15 valid submissions for Subtask 1, a promising outcome considering the time and resource constraints we had to prepare the task. Yet, our results make us optimistic about exploring future research avenues to enhance our system and study lighter approaches that can perform competitively. As future work, we aim to experiment with smaller and distilled models to encode textual, visual, and audio inputs, enabling us to fine-tune the full model cheaply.

Acknowledgments

This work has received supported by Grant GAP (PID2022-139308OA-I00) funded by MCIN/AEI/10.13039/501100011033/ and by ERDF, EU; the European Research Council (ERC), under the Horizon Europe research and innovation programme (SALSA, grant agreement No 101100615); Grant SCANNER-UDC (PID2020-113230RB-C21) funded by MICIU/AEI/10.13039/501100011033; Xunta de Galicia (ED431C 2020/11); by Ministry for Digital Transformation and Civil Service and "NextGenerationEU"/PRTR under grant TSI-100925-2023-1; and Centro de Investigación de Galicia "CITIC", funded by the Xunta de Galicia through the collaboration agreement between the Consellería de Cultura, Educación, Formación Profesional e Universidades and the Galician universities for the reinforcement of the research centres of the Galician University System (CIGUS).

References

Hassan Alhuzali and Sophia Ananiadou. 2021. SpanEmo: Casting multi-label emotion classification as span-prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1573–1584, Online. Association for Computational Linguistics.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. ViViT: A Video Vision Transformer.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ying Chen, Wenjun Hou, Shoushan Li, Caicong Wu, and Xiaoqiang Zhang. 2020. End-to-End Emotion-Cause Pair Extraction with Graph Convolutional Net-

work. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 198–207, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3161–3170, Online. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Timothy Dozat and Christopher D Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *International Conference on Learning Representations*.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Soumya Dutta and Sriram Ganapathy. 2024. HCAM – Hierarchical Cross Attention Model for Multi-modal Emotion Recognition.

Chuang Fan, Chaofa Yuan, Jiachen Du, Lin Gui, Min Yang, and Ruifeng Xu. 2020. Transition-based directed graph construction for emotion-cause pair extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3707–3717, Online. Association for Computational Linguistics.

Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. 2020. MEISD: A multimodal multi-label emotion, intensity and sentiment dialogue dataset for emotion recognition and sentiment analysis in conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4441–4453, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio Spectrogram Transformer.

Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. 2013. Challenges in Representation Learning: A report on three machine learning contests.

Tao Ji, Yuanbin Wu, and Man Lan. 2019. Graph-based dependency parsing with graph neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2475–2485, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization.

Fuyan Ma, Bin Sun, and Shutao Li. 2022. Facial Expression Recognition With Visual Transformers and Attentional Selective Fusion. *IEEE Transactions on Affective Computing*, 14(2):1236–1248.

André Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617–622, Sofia, Bulgaria. Association for Computational Linguistics.

Ryan McDonald. 2006. Discriminative training and spanning tree algorithms for dependency parsing. *University of Pennsylvania, PhD Thesis*.

Cam Van Thi Nguyen, Tuan Mai, Son The, Dang Kieu, and Duc-Trong Le. 2023. Conversation Understanding using Relational Temporal Graph Neural Networks with Auxiliary Cross-Modality Interaction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15154–15167, Singapore. Association for Computational Linguistics.

Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. 2022. Expanding Language-Image Pretrained Models for General Video Recognition.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling Relational Data with Graph Convolutional Networks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023a. Multimodal Emotion-Cause Pair Extraction in Conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. 2023b. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions.

Penghui Wei, Jiahao Zhao, and Wenji Mao. 2020. Effective inter-clause modeling for end-to-end emotion-cause pair extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3171–3181, Online. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

Wenjie Zheng, Jianfei Yu, Rui Xia, and Shijin Wang. 2023. A facial expression-aware multimodal multi-task learning framework for emotion recognition in multi-party conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15445–15459, Toronto, Canada. Association for Computational Linguistics.

A Appendix

Input	Description
U_i	Utterance i, defined as $U_i = (W_i, s_i, \varepsilon_i, \mathbf{X}_i, \mathbf{a}_i)$.
W_i	Word sequence of U_i as $W_i = (w_1, ..., w_{\ell\|w,i\|})$
s_i	Speaker of U_i, where $s_i \in \mathcal{S}$
ε_i	Emotion triggred in U_i, where $\varepsilon_i \in \mathcal{E}$
$\mathcal{S}$	Set of speakers in the dataset.
$\mathcal{E}$	Set of annotated emotions.
$\mathbf{X}_i$	Sequence of frames of U_i as $\mathbf{X}_i = (\mathbf{x}_1, ..., \mathbf{x}_{\ell\|x,i\|})$
$\mathbf{x}_k^{(i)}$	Specific frame of $\mathbf{X}_i$, where $\mathbf{x}_k^{(i)} \in \mathbb{R}^{h \times w \times 3}$.
$\mathbf{a}_i$	Sampled audio signal of U_i, where $\mathbf{a}_i \in \mathbb{R}^{\ell\|a,i\|}$.
$\ell\|w,i\|$	Length of the sequence W_i.
$\ell\|x,i\|$	Lenght of the sequence $\mathbf{X}_i$
Encoder	**Description**
$\mathbf{u}_i$	Encoder hidden representation of U_i from BERT, where $\mathbf{u}_i \in \mathbb{R}^{1024}$.
$\mathbf{W}_i$	BERT word embeddings of W_i as $\mathbf{W}_i = (\mathbf{u}_i, \mathbf{w}_1^{(i)}, ..., \mathbf{w}_{\ell\|w,i\|}^{(i)})$.
$\tilde{\mathbf{x}}_i$	Visual hidden representation for U_i, obtained as $\tilde{\mathbf{x}}_i = \text{LSTM}_\text{x}^{-1}(\text{ViT}(\mathbf{X}_i)) \in \mathbb{R}^{d_\text{V}}$.
$\tilde{\mathbf{a}}_i$	Audio hidden representation for U_i, obtained as $\tilde{\mathbf{a}}_i = \text{LSTM}_\text{a}^{-1}(\text{wav2vec}(\mathbf{a}_i)) \in \mathbb{R}^{d_\text{a}}$.
$\tilde{\mathbf{u}}_i$	Multimodal representation for U_i as $\tilde{\mathbf{u}}_i = (\mathbf{u}_i \| \tilde{\mathbf{x}}_i \| \tilde{\mathbf{a}}_i)$.
Decoder	**Description**
Φ	Arbitrary feed-forward network.
$\mathbf{c}_i$	Cause embedding for U_i as $\mathbf{c}_i = \Phi_c(\mathbf{u}_i) \in \mathbb{R}^{d_\text{G}}$.
$\mathbf{e}_i$	Effect embedding for U_i as $\mathbf{e}_i = \Phi_e(\mathbf{u}_i) \in \mathbb{R}^{d_\text{G}}$.
$\overline{\mathbf{c}}_i$	Emotion cause embedding for U_i as $\overline{\mathbf{c}}_i = \Phi_{c,\varepsilon}(\mathbf{u}_i) \in \mathbb{R}^{d_\text{G}}$.
$\overline{\mathbf{e}}_i$	Emotion effect embedding for U_i as $\overline{\mathbf{e}}_i = \Phi_{e,\varepsilon}(\mathbf{u}_i) \in \mathbb{R}^{d_\text{G}}$.
$\mathbf{C}$	Matrix of cause embeddings as $\mathbf{C} = (\mathbf{c}_1, ..., \mathbf{c}_m)$.
$\mathbf{E}$	Matrix of effect embeddings as $\mathbf{E} = (\mathbf{e}_1, ..., \mathbf{e}_m)$.
$\overline{\mathbf{C}}$	Matrix of emotion cause embeddings as $\overline{\mathbf{C}} = (\overline{\mathbf{c}}_1, ..., \overline{\mathbf{c}}_m)$.
$\overline{\mathbf{E}}$	Matrix of emotion cause embeddings as $\mathbf{E} = (\overline{\mathbf{e}}_1, ..., \overline{\mathbf{e}}_m)$.
$\mathcal{W}$	Trainable weights for the first biaffine module, where $\mathcal{W} \in \mathbb{R}^{d_\text{G} \times d_\text{G}}$
$\mathcal{W}_\varepsilon$	Trainable weights for the second biaffine module, where $\mathcal{W}_\varepsilon \in \mathbb{R}^{d_\text{G} \times \|\mathcal{E}\| \times d_\text{G}}$.

Table 2: Symbol notation.

NumDecoders at SemEval-2024 Task 7: FlanT5 and GPT enhanced with CoT for Numerical Reasoning

H. Andres Gonzalez Gongora[♣1*], **Md Zobaer Hossain**[♣2], **Jahedul Alam Junaed**[♠3]
[♣] University of Lorraine, Nancy, France
[♠] Shahjalal University of Science and Technology, Sylhet, Bangladesh
nanoandres_24@hotmail.com[1]
rowan.hossain@gmail.com[2]
jahedul25@student.sust.edu[3]

Abstract

In this paper we present a Chain-of-Thought enhanced solution for large language models, including flanT5 and GPT 3.5 Turbo, aimed at solving mathematical problems to fill in blanks from news headlines. Our approach builds on a data augmentation strategy that incorporates additional mathematical reasoning observations into the original dataset sourced from another mathematical corpus. Both automatic and manual annotations are applied to explicitly describe the reasoning steps required for models to reach the target answer. We employ an ensemble majority voting method to generate final predictions across our best-performing models. Our analysis reveals that while larger models trained with our enhanced dataset achieve significant gains (91% accuracy, ranking 5th on the NumEval Task 3 leaderboard), smaller models do not experience improvements and may even see a decrease in overall accuracy. We conclude that improving our automatic annotations via crowdsourcing methods can be a worthwhile endeavor to train larger models than the ones from this study to see the most accurate results.

1 Introduction

NumEval is a task first introduced in 2024 (Chen et al., 2024) building on previous work such as Cortis et al. (2017)'s fine-grained sentiment analysis (SemEval-2017 Task 5) and Jullien et al. (2023)'s clinical inference (SemEval-2023 Task 7). These prior tasks highlighted the importance of understanding numerical values in legal and medical contexts for determining outcomes. The primary objective of NumEval is to perform quantitative reasoning to generate numerical values corresponding to provided contexts.

In this project, we particularly focused on subtask 1 of task 3 (Huang et al., 2023) where our system must execute several mathematical calculations based on information from a provided passage to yield a numerical result used to fill in a headline with a blank. For instance, to complete the *CIA Cited Concerns About Snowden ____ Years Ago* headline, the model must subtract the article's publishing date by the explicitly stated date in the article (2009). Some entries involve a series of multiple mathematical operations that the model must perform.

Although numerical reasoning continues to present challenges to large language models (LLMs), advancements in larger models like DeepSeekMath (Shao et al., 2024) demonstrate promising capabilities in solving mathematical computations. DeepSeekMath is finetuned using different mathematical datasets and evaluated using Chain-of-Thought (CoT) prompting to provide intermediate reasoning steps. Inspired by CoT systems, we have developed a system pipeline that trains an encoder-decoder flanT5 (Chung et al., 2022) and an open source GPT 3.5 version[1] with additional mathematical corpora. These corpora include the Discrete Reasoning Over the Content of Paragraph (DROP) dataset (Dua et al., 2019) and another dataset which was manually and automatically annotated to include reasoning steps to reach the desired response. The core idea is that explicit intermediate reasoning, akin to chain-of-thought prompts, can enhance a model's quantitative reasoning capabilities (Wei et al., 2023).

In our revised approach, not only do we use smaller models ($\theta \leq 1$B)[2], but we also utilize multiple pipelines to determine the conditions under which our model achieves the highest accuracy. Firstly, we establish a baseline by fine-tuning with the provided dataset (Huang et al., 2023), then we incorporate additional observations from the DROP dataset into our training data. Thirdly, we adopt

[*] All authors have equal contributions

[1] List of open source OpenAI GPT models

[2] θ refers to model parameters

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1260–1268

a Chain-of-Thought (CoT) approach, fine-tuning both a flanT5 model and a generative open-source OpenAI model (GPT 3.5 Turbo) with more detailed inputs and outputs, including string normalizations and quantitative reasoning steps. Finally, we employ an ensemble majority voting method to select the best results from these models, resulting in a 91% accuracy and 5th place on the leaderboard of the NumEval competition [3].

2 Related Works

Through pre-training on a vast amount of text data, LLMs can develop a broad knowledge base encompassing numerical concepts, arithmetic operations, and mathematical relationships. Lewkowycz et al. (2022) propose a language model named Minerva, which demonstrates strong performance on various quantitative reasoning tasks, including undergraduate-level physics or chemistry problems.

Numerical reasoning has been extensively studied across diverse contexts, including word embedding (Wallace et al., 2019; Naik et al., 2019; Sundararaman et al., 2020) and math word problems (Wang et al., 2018; Cobbe et al., 2021). Within the domain of Question Answering, several approaches have been proposed. Xu et al. (2022) present a framework called Diagnosing Numerical Capabilities (DNC), which involves two stages: recognition of numbers in the context and question to treat them as candidate operands, followed by the correct selection of operands and operations based on understanding questions and context. Kim et al. (2022) proposes an attention-masked reasoning model that learns to leverage the number-related context to alleviate the over-reliance on parametric knowledge and enhance the numerical reasoning capabilities of the QA model. Other studies, such as those by Geva et al. (2020) and Feng et al. (2021), explore the infusion of external knowledge to augment the numerical reasoning skills of the models. Yang et al. (2021) focus on Numerical Reasoning over Text (NRoT) using T5 models, employing five training pipelines and multitasking training to progressively enhance model performance through tasks such as general reading comprehension and fine-tuning on the DROP dataset (Dua et al., 2019). Additionally, in reasoning tasks, Chain-of-Thought prompting has shown promise in improving the performance of large language models (Ling et al., 2024). While Chain-of-Thought (CoT) allows models to generate more comprehensive reasoning processes, it also introduces challenges such as hallucinations and accumulated errors. To mitigate these issues, the authors propose enabling explicit and deductive rigorous reasoning within language models. They emphasize the importance of self-verification for trustworthiness, which leads to significantly improved answer correctness in reasoning tasks. Drawing inspiration from these CoT-based methods, we incorporate them into our approach due to their superior performance in numerical reasoning tasks.

3 System Description

In our system, we defined three main pipelines that were compared against a baseline encoder-decoder model. Specifically, we used an instruction finetune model version (flan) of the Text-To-Text Transfer Transformer (T5) (Chung et al., 2022). This flanT5 model underwent fine-tuning in its small, base, and large versions, employing a learning rate of 5e-5 for 5 epochs and a batch size of 2.

3.1 DROP Dataset

To enhance performance beyond the baseline, we merged the Discrete Reasoning Over the Content of Paragraph (DROP) dataset (Dua et al., 2019) with the original numerical headline generation dataset (Huang et al., 2023). The DROP dataset consists of paragraphs with answer spans to given questions, often referencing multiple positions in the provided passage. With a total of 77400 observations in the training data split, we filtered out 46973 observations related to numerical reasoning tasks. Due to computational constraints, we merged only 20000 of these filtered observations with the original headline generation dataset. The selection of these 20,000 entries was based on a random seed of 43. Additionally, it's important to note that while the input text in the DROP dataset is structured as questions, unlike the fill-in-the-blank format used Huang et al. (2023)'s dataset, we transformed the questions into masked headlines by locating the answer in the original dataset and masking it from the passage's headline.

3.2 GPT 3.5 turbo

For this task, we utilized the GPT 3.5 Turbo model to extract numerical reasoning and explanations from the NumHG dataset (Huang et al., 2023).

[3] GitHub repository for our system

Prompt selection plays a critical role in obtaining optimal output from the GPT model. White et al. (2023) outline various prompt engineering techniques in a pattern-based catalog that have been successfully applied to improve the outputs of large language models (LLMs) in conversations. Drawing from the insights provided by White et al. (2023), we adopt three distinct patterns into our prompt design: the Persona Pattern, the Context Manager Pattern, and the Recipe Pattern. Each pattern was carefully selected to address specific challenges and enhance the interpretability of the generated responses.

Persona Pattern: It assists the GPT model in determining the types of output to generate and which details to prioritize. By incorporating persona-based prompts, we guide the model to discern the essential information to emphasize in its responses.

Context Manager Pattern: The goal of this pattern is to focus on specific topics and exclude unrelated ones from consideration. Through careful manipulation of contextual cues, we enhance the model's ability to generate contextually relevant and coherent numerical explanations.

Recipe Pattern: It introduces constraints to ultimately output a sequence of steps based on partially provided "ingredients" required to achieve a specified goal. Serving as a structured framework for our prompt design, the Recipe Pattern guides the model in constructing step-by-step sequences.

Role	Content	Matched Pattern
System	You are a helpful assistant, skilled in providing numerical reasoning.	Persona Pattern
User	context: [news] + [masked headline]	–
User	The answer to the fill-in-the-blank question is [ans]. Please provide a complete sequence of numerical reasoning steps in a paragraph format that is used to derive this answer. Begin your response by discussing the relevant sentences, and then outline the numerical reasoning steps. Conclude your response with: 'So the answer is [ans].'	Context Manager & Recipe Patterns

Table 1: Conversation prompt with matched patterns. Here, placeholder values are from the dataset.

3.3 Chain of Thought (CoT)

To further steer the capabilities of both the decoder GPT 3.5 Turbo and our trained flanT5, we incorporated chain-of-thought (CoT) prompting (Wei et al., 2023). This involved adding specific reasoning steps in the output text that the model relied on to produce the numerical response. According to Wei et al. (2023), CoT outperforms traditional prompting and finetuning approaches by providing intermediate reasoning steps that facilitate model interpretation. Moreover, in large models, even a few CoT sequences can outperform some finetuned pre-trained models in arithmetic and symbolic reasoning tasks (Wei et al., 2023).

In our CoT pipeline, our initial approach involved an automatic annotation step, which we supplemented with manual annotation to handle more complex calculations. Below, we outline this annotation process, including additional preprocessing steps implemented to normalize the input and output data.

Automated Annotation: In the original news articles, dates are written in abbreviated form and placed within brackets before the passage. Since many headline completion tasks involve subtracting a given number of years mentioned in the article from the publishing date, we extract this metadata date and transform it to prefix the overall passage with a descriptive sentence. For instance, an article with the date *(Feb 13, 2013 6:54 PM)* is transformed to *The news was published on 13th February in the year of 2013*. This approach enables our models to retrieve explicit and normalized dates for performing the corresponding mathematical operations.

Answer extraction is conducted using the *spacy* module to tokenize each passage and iterate over each resulting sentence with a custom placeholder function. If the answer is found within a sentence, it is extracted. The main answer extraction function is then applied to our main 7 placeholder functions to automate the annotation of the simpler calculations. Among these, 5 (*copy, translation, round, sround, and paraphrase* (Huang et al., 2023)) are much more straightforward, whereas *subtract* and *span* require a heuristic-based annotation, where the answer string is preprocessed to fit the appropriate format.

For instance in our span recipe, (*get_span_placeholder*), we modify the resulting string if the blank contains the following tokens.

- **No._** which we pass to the model as output with the following explanation: *No. 1 typically refers to the topmost or the best-ranked item in a list or a competition.*

- **_M** which we pass to the model as output with the following explanation: *The letter 'M' in the headline indicates that the answer refers*

to an amount that should be transformed to millions

- **_st** which we pass to the model as output with the following explanation: *The presence of 'st' in the headline gives a clue that the answer is 1*

Otherwise, we specify that the span containing the answer may refer to a person, object or event.

As previously stated, each of the simple aforementioned calculations has its own placeholder function, which we further examine in Table 6 and pass to our main algorithm in Figure 1.

One of the biggest challenges the automation system faced was inconsistent annotations from the original dataset wherein certain passages would not contain references to the answers at all or, more egregiously, wrong calculations. For the headline *Wife who got $1B in Divorce: Not Enough* where 1 corresponds to the answer, the calculation is as follows *Round(Paraphrase(995, K), 0)*. Nevertheless, the paraphrase should have an *M* instead of a K as the value is given in the millions rather than the thousands.

Furthermore, apart from the provided calculation, the passages often lack explicit numerical reasoning to justify why a model should yield the floor value of a decimal number for a headline instead of rounding it up. For example, in the article *Woman Places $615K Bet on Hillary Clinton* the value must be paraphrased and then rounded up to the nearest whole number to reach the answer of 615. However, the passage states that “"a 46-year-old woman just placed a $615,862 bet on Clinton". Mathematically, the number should be converted to thousands by dividing by 1000 and then rounded up, resulting in an answer of 616. Notwithstanding, the headline reports 615.

Manual Annotation: We employed manual annotation to address more complex operations, including addition, subtraction, multiplication, and division. In each case, we began with an automated step using GPT 3.5, as described in Table 1 and then manually cleaned up the reasoning steps, as well as, overall responses using a frontend system built with *streamlit*. Figure 2illustrates an example where we manually corrected the automated annotation to describe the steps for solving both simple and more complex calculations in a fill-in-the-blank question. In some instances, answers were incorrect, or the original logic provided by the model was overly redundant or incorrect. Consequently, we relied on 3 main human annotators[4] to review the 1K annotations completed by GPT 3.5 turbo. In Table 7, we can see some examples of patterns annotators followed to make sure the dataset would be consistent.

With both our automatic and manual annotations combined, we proceeded to fine-tune our GPT 3.5 Turbo and flanT5 models to evaluate whether the improved dataset yielded any advantages over the baseline. For this fine-tuning process, we maintained the same hyperparameters as before, except for the batch size. The batch size was increased for the small and base-sized flanT5 models to 16 and 8, respectively. This adjustment was necessary because we trained these models using a larger GPU, an A100 40GB GPU.

3.4 Ensemble

Lastly, we implemented an ensembling method using majority voting, wherein for each passage, we selected the numerical answer with the most votes as the correct one. In our ensembling pipeline, we narrowed down our majority voting to 4 models, consisting of our best-performing models: one version of large flanT5 trained for 3 epochs using only the NumHG dataset, another large flanT5 trained using NumHG for 2 epochs, a flanT5 trained for 2 epochs using the DROP dataset, and a CoT fine-tuned version of GPT 3.5 Turbo. We included versions that were trained for 2 epochs instead of 3 as they outperformed their 3-epoch counterparts, particularly the DROP-trained flanT5. However, this was only the case with the large models, as the base and small ones consistently performed better after training for 3 epochs rather than 2. During the evaluation period, we were unable to finish training the CoT models; therefore, we only used the available top 4 models for ensembling.

Since we employed an even number of models for this method, the likelihood of encountering ties is high. In instances of a tie, where a unanimous answer majority was absent, we resorted to the answer generated by our top-performing model—FlanT5 fine-tuned exclusively with NumHG.

4 Results

Based on the results presented in Tables 2 and 3, we observe that the difference in performance between the small and base flanT5 models is not par-

[4]These annotators are the authors of this paper

```
def get_ans_sent(item):

   operations = {"Copy":get_copy_placeholder,"Trans":get_trans_placeholder,
              "Span":get_span_placeholder,"Round":get_round_placeholder,"Paraphrase":
                  ↪ get_paraphrase_placeholder,
              "Subtract":get_subtract_placeholder, "SRound":get_round_placeholder}

   for operation, function in operations.items():

      if check_calculation(item, operation):

         return function(item)

   return f"So␣the␣answer␣is␣{item['ans']}"
```

Figure 1: Main function used to annotate our data automatically. Each placeholder contains the find answer function, which tracks the main spans needed to fill in the blank question.

	T5 Flan Small	T5 Flan Base	T5 Flan Large
NumHG	0.83	0.89	0.91
NumHG+DROP	0.84	0.88	0.90
COT	0.58	0.83	0.88

Table 2: Results of T5 Flan models trained on three different datasets with the validation set.

	T5 Flan Small	T5 Flan Base	T5 Flan Large
NumHG	0.82	0.84	0.90
NumHG+DROP	0.83	0.88	0.90
COT	0.58	0.83	0.88

Table 3: Results of T5 Flan models trained on three different datasets with the test set.

ticularly notable, except when employing the CoT method, where the small models significantly underperform. Additionally, as shown in Table 4, it is surprising to note that a finetuned GPT 3.5 Turbo model underperforms compared to the other flanT5 models, despite its larger size. Overall, our team ranked 5th out of 16 teams, including the baseline, on the final leaderboard, achieving 91% accuracy with our majority model.

	dev	test
NumHG	0.91	0.90
NumHG+DROP	0.90	0.90
COT	0.88	0.88
GPT 3.5	0.85	0.84
GPT 3.5 (fine tuned)	0.81	0.82
Ensemble (Majority)	0.92	0.91

Table 4: Best Results of the models on validation and test set.

5 Discussion

Our CoT results, as observed in Tables 2 and 3 align with the findings reported by Wei et al. (2023), indicating that smaller models do not experience significant gains when using prompting, partly due to their fewer parameters. In their study, it is explicitly mentioned that models in the range of 100 billion parameters or more exhibit the highest gains. However, all of the flanT5 models we utilized have significantly fewer parameters, failing to reach the 1 billion mark (Chung et al., 2022). We believe that conducting CoT experiments with the XL and XXL versions of these models would likely result in much more significant improvements.

5.1 Error Analysis

For our error analysis, we converted our model predictions into strings to facilitate comparison with their corresponding ground truths. It's important to note that while the competition required numerical values to be uploaded, some ground truths were formatted with commas (e.g., 1,500 instead of 1.5) or included important dates such as 9/11. In cases like the latter, where the ground truth couldn't be converted to a real number, we cast our results to string values. However, even with this adjustment, discrepancies in formatting, such as our model yielding 4.5 while the ground truth is 4.50, resulted in evaluations as incorrect. When accounting for these differences, the accuracy rate of the majority voting ensembling method reached 93%. Additionally, some answers in the test set were tagged as unanswerable.

In Table 5, we observe the error rate of our best majority voting ensembling method. Despite our best-performing model achieving a 91% accuracy rate, as noted in Table 4, we can see a high error rate for complex operations such as addition, multi-

ply, and subtraction. Additionally, the surprisingly high error rate for the round operation may stem from inconsistencies in the annotation process. As mentioned in Section 3, there are no contextual hints in the passage besides the calculation to aid the model in flooring a value instead of rounding it up. Moreover, certain calculations that instruct the model to round up a value, such as *2.8*, have a ground truth of 2 instead of 3.

Nevertheless, our models encountered several round-up errors where they failed to generalize properly, particularly when rounding up to the nearest tenth. For example, in an operation yielding 4.831 where rounding up to the nearest tenth should result in 4.83, our models rounded it up to 4.8. Similarly, in cases where 4.8 should be rounded up to the nearest whole number, our flanT5 models often failed to round it up to 5, opting instead for 4. In approximately 80% of cases where round operations were inaccurately predicted, the primary issue was the selection of an incorrect upper or lower bound for the rounding operation.Many of these mistakes involved multiple complex calculations, where a round operation had to be computed after 2 or 3 additional computations. An example of this issue can be seen in the operation: *Round(Divide(85,12),0)* where the result is supposed to be 7, but the model incorrectly yields 85 to complete the headline *Robert Durst Gets ____ Years for Gun Charges*. However, the article explicitly states “"Robert Durst, millionaire oddball and star of HBO’s true-crime documentary The Jinx, pleaded guilty to gun charges Wednesday in New Orleans, earning him 85 months in prison". While the headline requests years, the model fails to convert the value in months to years by dividing by 12, instead simply copying the number *85* from the span.

Similarly, we encountered errors with the copy operation, where either the model would copy an incorrect value or, more egregiously, round it up to another value. For instance, in the headline *Spanish Bank Offers $____B to Madoff Victims* the article states that “Spanish banking giant Banco Santander, whose clients lost nearly $3.1 billion in Bernard Madoff’s Ponzi scheme, has offered to pay back customers some $1.82 billion, reports Bloomberg.". Therefore, the correct answer should be 1.82, but our model incorrectly rounds this value to 1.8. We estimate that if we account for these errors, our majority ensembling could potentially achieve a 2% increase over our 91% result.

Finally, the divide calculation presents fewer errors in our models, with most mistakes occurring within complex operations involving multiple calculations. However, it’s worth noting that ratio conversion, specifically converting a ratio to a percentage and viceversa, poses a challenge for our models. This challenge is evident in the *Divide(1,20%)* calculation where the expected result is 5 to complete the headline *Odds of a Depression? 1 in ____*. Unfortunately, our model yields 4, indicating that it interpreted 20% as 25%. While such corner cases underscore that our models may not always accurately translate fractions to their corresponding real numbers (i.e., dividing the percentage number by 100), it’s important to consider the context. In the article, multiple percentages were mentioned in the following spans: “The bad news is that this recession is likely to be America’s worst since WWII—but the good news is there’s only a 20% chance it will become a depression (...) The lack of any major global conflicts means the chance of a depression being a major one—a decline of 25% or more—is only 2%". In other words, the model did not identify the span with the right percentage to convert to its corresponding ratio, which is 20% rather than 25%.

6 Conclusion

In this paper we observed that introducing additional observations with detailed reasoning steps can enhance a model’s ability to solve mathematical problems while also highlighting areas where its reasoning may fall short. Nevertheless, our results suggest that larger models may derive the most benefit from a CoT + finetuning approach. Because of this, we argue that leveraging larger LLMs could lead to even greater gains in quantitative reasoning tasks. Furthermore, while we have provided access to our open dataset[5], we recognize the importance of improving automatic annotations through crowdsourcing to achieve more accurate results. Given that our automatic annotations sometimes exhibit issues in reasoning steps compared to manual annotations, and certain entries in the original dataset are ambiguous or erroneous, we emphasize the necessity of data cleanup to enhance mathematical reasoning in language models.

[5]COT Automatic and Manually annotated dataset

References

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. SemEval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 519–535, Vancouver, Canada. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Yu Feng, Jing Zhang, Xiaokang Zhang, Lemao Liu, Cuiping Li, and Hong Chen. 2021. Injecting numerical reasoning skills into knowledge base question answering models. *arXiv preprint arXiv:2112.06109*.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. *arXiv preprint arXiv:2004.04487*.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Jeonghwan Kim, Junmo Kang, Kyung-min Kim, Giwon Hong, and Sung-Hyon Myaeng. 2022. Exploiting numerical-contextual knowledge to improve numerical reasoning in question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1811–1821, Seattle, United States. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2024. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36.

Aakanksha Naik, Abhilasha Ravichander, Carolyn Rose, and Eduard Hovy. 2019. Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3374–3380.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Dhanasekar Sundararaman, Shijing Si, Vivek Subramanian, Guoyin Wang, Devamanyu Hazarika, and Lawrence Carin. 2020. Methods for numeracy-preserving word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4742–4753.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. *arXiv preprint arXiv:1909.07940*.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating a math word problem to an expression tree. *arXiv preprint arXiv:1811.05632*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Jialiang Xu, Mengyu Zhou, Xinyi He, Shi Han, and Dongmei Zhang. 2022. Towards robust numerical question answering: Diagnosing numerical capabilities of nlp systems. *arXiv preprint arXiv:2211.07455*.

Peng-Jian Yang, Ying Ting Chen, Yuechan Chen, and Daniel Cer. 2021. Nt5?! training t5 to perform numerical reasoning. *arXiv preprint arXiv:2104.07307*.

A Error Analysis

	Percentage Error Rate	Count Error Rate
Addition	46%	42
Copy	4%	148
Divide	36%	4
Multiply	74%	20
Paraphrase	4%	16
Round	55%	101
Span	7%	1
Subtract	63%	59
Translation	2%	20

Table 5: The error rates are over the total amount of a given operation, not for the whole dataset. Note that Sround error rate is computed with the round operation.

B Data Annotation

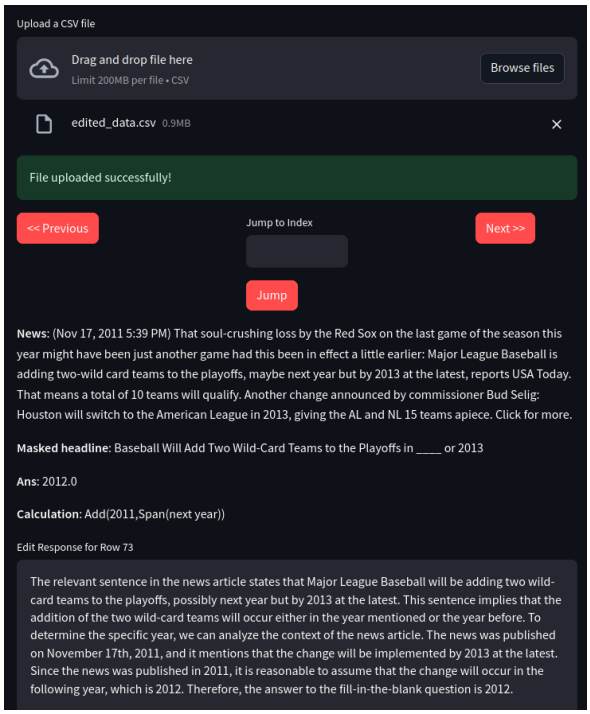


Figure 2: Example of our manual annotation system on streamlit

Recipe	Function	Example	Operation
Copy	The simplest placeholder as the model simply takes the exact response taken from a given span in the passage	A union repping 2 million health care workers has made quite a find: 39 million N95 masks	*39* → 39
Translation	Similarly to the copy placeholder, the answer is present in a given sentence. Thus, we state that the answer must be converted to its corresponding numerical value.	A University of Utah student paid his tuition bill with 2,000 one-dollar bills	*one* → 1
Paraphrase	Involves paraphrasing a value that is appended in the headline by *K*, *M*, or *B*. That is to say, if the value is to be expressed in the thousands (K), millions (M), or billions (B), the numerical value found in the passage must be transformed accordingly.	A Florida travel insurance company has awarded a Georgia high school teacher $10,000	*$10,000* → 10
Round	Akin to paraphrase, round implies rounding a value to its nearest whole number or tenth depending on the specified decimal in the calculation.	Hackers made public the email addresses, usernames, and passwords of 790,724 Brazzers members.	*$790,724* → 791
Sround	Instead of approximating to the greater value, in *sround* the model must transform the value to its nearest floor value.	Today's after-hours bad news from the credit-crunch front comes from insurer AIG, which reported a fourth-quarter loss of $5.29 billion	*$5.29* → 5
Span	It fetches the span in the given passage the headline is referring to	Brooklyn store owner Jacob Hamula could have ended up a victim of Salvatore Perrone, the suspected serial killer believed to have gunned down three other store owners before police nabbed him.	*Brooklyn store owner Jacob Hamula* → 1
Subtract	It implements a heuristic whereby the model subtracts between the published date and the date mentioned in the passage as long as these dates are present in the metadata date and the article	(Apr 1, 2014 4:03 AM CDT) Steve Jobs did it; Google founders Sergey Brin and Larry Page did, too. Now Mark Zuckerberg is joining the ranks of the $1-a-year CEOs, Bloomberg reports. That's what the Facebook boss earned in salary last year	*(Apr 1, 2014 4:03 AM CDT)* & *last year* → 2014 - 1 = 2013

Table 6: Placeholder functions used in our automatic annotation. Note that the round operation includes a paraphrase one in the given example. Additionally, the recipe for round and sround is virtually the same. Finally, the only subtract operations that were automatically annotated with this method involve dates. Otherwise, they are deemed as more "complex" operations that were manually annotated.

Recipe	Pattern Example
Paraphrase	From the presence of "M" at the end of the fill-in-the-blank, we can infer that the blank in the question is asking for the value in millions. The sentence states that the population will be 308,400,408, so we need to convert this value to millions. To do this, we divide 308,400,408 by 1,000,000 which gives us 308.400. Since the question asks for the value in millions, we round down to the nearest whole number, which is 308. So the answer is 308.
Translation (transform dates)	The presence of both the apostrophe (') and "s" surrounding the blank strongly indicates that the number is abbreviated and pertains to a decade. Taking the example of the '80s, which covers the years 1980 to 1989, when individuals refer to "the '80s," they are typically referring to the complete decade. Since 1987 is within the timeframe of the '80s, it logically follows that the appropriate response is 80. So the answer is 80.
Translation (transform to ratio)	The term "quarter" refers to one part out of four equal parts. In the context of numbers or fractions, "1 in 4" is used to express the concept of a quarter. This means that when something is divided into four equal parts, you are referring to one of those parts.

Table 7: Some example patterns used by the annotators to keep annotations consistent across some example tasks.

FZI-WIM at SemEval-2024 Task 2: Self-Consistent CoT for Complex NLI in Biomedical Domain

Jin Liu[1, 2] and Steffen Thoma[1]

[1]FZI Research Center for Information Technology, Karlsruhe, Germany
[2]Karlsruhe Institute of Technology, Karlsruhe, Germany
{jin.liu, thoma}@fzi.de

Abstract

This paper describes the inference system of FZI-WIM at the SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. Our system utilizes the chain of thought (CoT) paradigm to tackle this complex reasoning problem and further improves the CoT performance with self-consistency. Instead of greedy decoding, we sample multiple reasoning chains with the same prompt and make the final verification with majority voting. The self-consistent CoT system achieves a baseline F1 score of 0.80 (1st), faithfulness score of 0.90 (3rd), and consistency score of 0.73 (12th). We release the code and data publicly[1].

1 Introduction

The Safe Biomedical Natural Language Inference for Clinical Trials (NLI4CT) task aims to investigate the consistency and faithfulness of natural language inference (NLI) models in clinical settings(Jullien et al., 2024). NLI is a typical natural language task requiring natural language reasoning (Yu et al., 2023). Fine-tuned BERT-based (Devlin et al., 2019) discriminative models have been widely applied to solve NLI problems (Liu et al., 2020; He et al., 2021). Studies show increasing reasoning capabilities of large language models (LLMs), both in proprietary (Brown et al., 2020; Achiam et al., 2023) and open-source LLMs (Touvron et al., 2023a,b; Jiang et al., 2024). However, problems of inconsistency and unfaithfulness still occur with LLMs(Golovneva et al., 2023; Turpin et al., 2023). Compared to other domains, medical applications have much higher standards regarding safety and trustfulness, so inconsistencies and unfaithfulness limit AI applications in the medical domain.

Chain of Thought (CoT) has been proposed to elicit the reasoning capabilities of LLMs (Wei et al., 2022). Based on the CoT, further concepts like Tree of Thought (Yao et al., 2023a), ReAct (Yao et al., 2023b), Self-Consistency (Wang et al., 2023), and so on have been proposed to improve the performance of CoT further. One common characteristic of the frameworks mentioned above is the explicit generation of reasoning chains. Since only verification labels are provided in the NLI4CT training dataset, we utilize GPT-4 to generate reasoning chains. With the distilled knowledge from GPT-4, we further instruction-tune an open-source LLM with low-rank adaption (LoRA) (Hu et al., 2022) for claim verification with CoT. Our system follows the self-consistency concept by generating multiple CoT reasoning chains and verifying with majority voting.

We summarize our major findings regarding this task as follows:

- LoRA instruction-tuning can bring domain-specific (biomedical) knowledge and reasoning capabilities to LLMs.
- Our instruction-tuned LLM tends to contradict the statement if the information is only contained in the statement, not in the premise, even if the information is factually correct.
- CoT reasoning gains significant performance improvement regarding faithfulness compared to the label-only prediction.
- Compared to the greedy CoT, self-consistent CoT with majority voting has a performance improvement of 1.31 (baseline F1), 0.75 (consistency score), and 0.69 (faithfulness score) percentage points. Performance improvement is limited for the binary classification problem.

2 Background

NLI4CT contains one text entailment task, namely infer the relationship between a premise and a state-

[1]https://github.com/jens5588/FZI-WIM-NLI4CT

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1269–1279

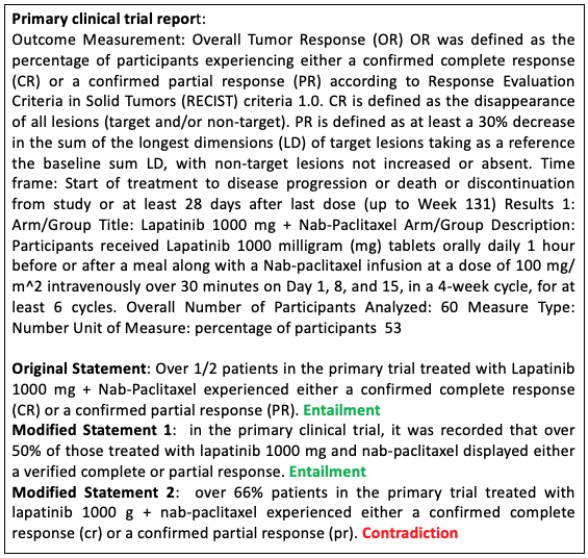


Figure 1: A data example. With the same clinical report, semantic-preserving and semantic-altering interventions on the original statement are used to evaluate the consistency and faithfulness of the verification system.

ment as either entailment or contradiction (Jullien et al., 2024). The premises in the NLI4CT dataset are collected from publicly available breast cancer clinical trial reports (CTRs), split into four sections: eligibility criteria, intervention, results, and adverse event (Jullien et al., 2023). The statements are sentences making claims about the information in the CTR premise, either about a single CTR or a comparison between 2 CTRs. The numbers of training, validation, and test examples are 1700, 200, and 5500, respectively. In the test set, 500 examples are used as anchors. The other 5000 statements are created with interventions on these first 500 examples. Figure 1 shows an example with interventions on the statements. Using the same clinical report, two statements are modified, one semantics-preserving (modified statement 1) and the other semantics-altering (modified statement 2), based on the original statement. The purpose of the interventions is to investigate the consistency and faithfulness of the inference.

For the text entailment task in the first iteration of NLI4CT (SemEval-2023 Task 7), most systems have fine-tuned discriminative transformer-based models (Jullien et al., 2023). Kanakarajan and Sankarasubbu (2023) instruction-tuned Flan-T5 model and achieved the 2nd place for the text entailment task. However, the systems mentioned above only predicted the verification labels without the reasoning process. To achieve a trustworthy verification, our system not only predicts the label but also the reasoning chains. Since the training and validation datasets have only provided verification labels. We utilize GPT-4 to verify with reasoning chains for training and validation datasets. We further instruction-tune an open-source LLM with distilled reasoning chains from GPT-4. To address the inconsistency problem in reasoning chains, we sample multiple chains and then employ a majority voting approach to determine the final verification outcome.

3 System overview

Figure 2 shows the pipelines of data creation, model training, and inference of our system. In the following, we describe each part.

3.1 Knowledge Distillation

The NLI4CT data only contains verification labels without rationales. Following the concept of knowledge distillation (Hinton et al., 2015), we leverage GPT-4 to generate rationales with CoT for the training and validation datasets. We further extract the verification labels of GPT-4 based on the CoT reasoning and filter out the examples for which the extracted labels are different from the gold labels.

3.2 LoRA Instruction-tuning

Parameter efficient fine-tuning (PEFT) gains popularity as the sizes of LLMs increase (Houlsby et al., 2019; Pfeiffer et al., 2020; Li and Liang, 2021; Hu et al., 2022). Low-rank adaption (LoRA) (Hu et al., 2022) is a PEFT approach to fine-tune LLMs. Studies show over-parametrized models reside on a low intrinsic dimension (Aghajanyan et al., 2021; Li et al., 2018; Hu et al., 2022). The key assumption of LoRA is that the updates to the weights also have a low intrinsic rank during adaptation for downstream tasks. The parameter updates of a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$ can be represented as $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$ and $r \ll \min(d, k)$ (Hu et al., 2022).

Based on the concept of LoRA, given prompt x and the target output $y = (y_1, ..., y_m)$, the loss can be formulated as:

$$L = \sum_{i=1}^{m} -\log(p_\theta(\hat{y}_i = y_i|x, y_1, ..., y_{i-1})), \quad (1)$$

where θ represents W_0, B, A and only B and A are trainable.

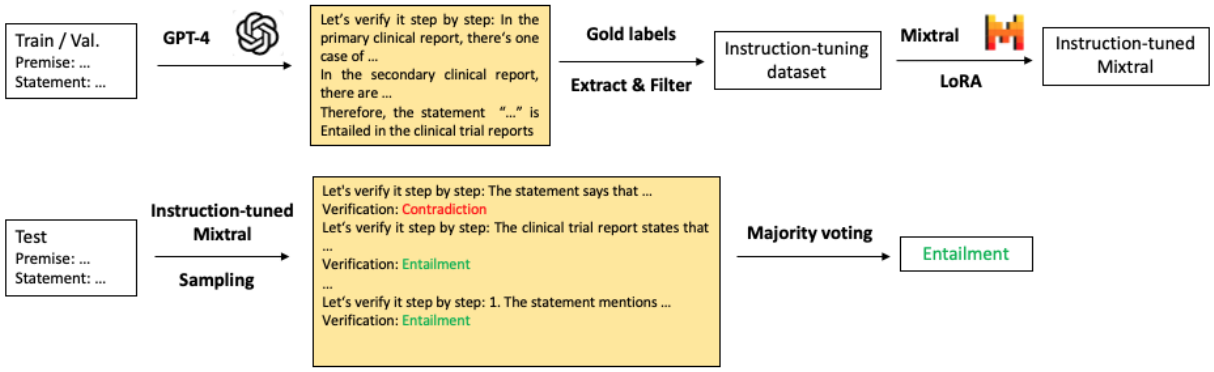


Figure 2: Training and Inference pipeline of self-consistent CoT system.

3.3 Self-Consistency

Self-consistency for LLMs was proposed by Wang et al. (2023) to replace the greedy decoding in CoT reasoning. The intuition behind self-consistency is that there are multiple ways to solve a complex problem. Another fact that supports the introduction of the self-consistency concept is that current LLMs still have difficulties with complex reasoning tasks and make mistakes in certain reasoning steps. With multiple reasoning chains, the model is less likely to make the same error and reach the same wrong answer. Wang et al. (2023) show self-consistency boosts the performance of different LLMs with different sampling strategies, including temperature sampling, top-k sampling, and nucleus sampling, on arithmetic, commonsense and symbolic reasoning tasks.

4 Experimental setup

In this section, we describe our experiment setup. We describe the CoT generation with GPT-4, the model training, and the inference setups.

4.1 LoRA Instruction-tuning

Instruction Data Creation Since GPT-4 has state-of-the-art reasoning capabilities, we instruct GPT-4 to verify the statements based on the CTR premise step-by-step for the training and validation datasets. The instruction prompt is shown in Appendix A.1. We then extract the verification labels of CoT rationales with an NLI model, namely bart-large-mnli (Lewis et al., 2020), and keyword matching. We compare the extracted labels with gold labels and filter out examples where the verification is wrong. For 1700 training and 200 validation examples, we achieve 1413 and 166 correctly verified examples with CoT, respectively, resulting in an accuracy of approximately 83%. With these selected examples, we build an instruction-tuning dataset for fine-tuning an open-source LLM. We show an example of CoT instruction-tuning data in Appendix A.2. For further comparison, we create another instruction dataset containing only verification labels without CoT for training and validation datasets. Appendix A.3 shows one label-only instruction-tuning data example.

Instruction-tuning We select Mixtral-8x7B-Instruct (Jiang et al., 2024) as our open-source LLM for instruction-tuning. The base model selection has considered reasoning capabilities and the number of model parameters. The configuration of LoRA is set as follows: r=8, alpha=32 and adaption of the attention weights of query (W_q), key (W_k), value (W_v) and output (W_o). This setup leads to 6.8M trainable parameters, which corresponds to 0.0146% of the base model size of 46.7B. We instruction-tune Mixtral-8x7-Instruct with the CoT instruction dataset and label-only instruction data separately for five epochs. Further implementation details are described in Appendix B.1.

4.2 Inference

Self-Consistent Inference To generate reasoning chains with the CoT instruction-tuned Mixtral model, we follow Wang et al. (2023) using a temperature sampling with the parameters $T = 0.7$ and $k = 50$ in top-k. For each (premise, statement) pair in the test dataset, we begin with sampling 10 reasoning chains. We then extract the verification labels and apply majority voting to decide the final label. If the result for both labels is tied, we further generate reasoning chains with different seeds. The maximal number of generated reasoning chains is 25. We show the number of generated reasoning chains in Appendix B.2.

Greedy Inference For comparison, we also apply the greedy decoding strategy to generate verification with label-only and CoT instruction-tuned models. With the greedy strategy, the models predict the next token with the highest probability without sampling[2].

4.3 Evaluation Metrics

Three evaluation metrics are used for the task: baseline F1, faithfulness score, and consistency score. The F1 score is used to evaluate the performance of the control set without intervention, consisting of 500 samples. There are 5000 samples in the test dataset, created with interventions based on the samples in the control set.

The consistency score evaluates whether the model predicts the same label when semantic-preserving interventions exist in the original statements (Jullien et al., 2024). For N statements x_i in the contrast set C, their corresponding original statements y_i and model prediction function f, the consistency score is calculated with:

$$Consistency = \frac{1}{N}\sum_{1}^{N} 1 - |f(y_i) - f(x_i)| \quad (2)$$

where $x_i \in C$: $\text{Label}(x_i) = \text{Label}(y_i)$.

The faithfulness score evaluates whether the model can change its prediction when semantic-altering interventions are present in the original statements (Jullien et al., 2024). For N statements x_i in the contrast set C, their corresponding original statements y_i and model prediction function f, the faithfulness score is calculated with:

$$Faithfulness = \frac{1}{N}\sum_{1}^{N} |f(y_i) - f(x_i)| \quad (3)$$

where $x_i \in C$: $\text{Label}(x_i) \neq \text{Label}(y_i)$, and $f(y_i) = \text{Label}(y_i)$.

5 Evaluation

In this section, we report the results of our self-consistent CoT system. For comparison, we also report the results of label-only verification and CoT-greedy. The comparison of three systems serves as an ablation study to analyze the performance improvement of extra steps, reasoning chain generation and self-consistent verification. Table 1 gives an overview of the three systems compared to the best scores in each category. Our self-consistent CoT system achieves 1st place regarding baseline F1, 3rd place regarding faithfulness, and 12th place regarding consistency. Next, we will analyze our system according to the three metrics separately.

[2] https://huggingface.co/docs/transformers/generation_strategies

Model	Base F1	Consistency	Faithfulness
Label-only Greedy	0.7867	**0.7364**	0.8102
CoT Greedy	0.7869	0.7217	0.8970
Self-Consistent CoT	**0.8000**	0.7292	**0.9039**
Best score	0.80	0.81	0.95

Table 1: Overview of three systems compared to the best scores of the task regarding each metric.

Model	Eligibility	Intervention	Adverse Events	Results
Label-only Greedy	0.7482	**0.8175**	0.8131	0.7679
CoT Greedy	0.7132	0.7794	**0.8667**	0.7961
Self-Consistent CoT	**0.7581**	0.7770	0.8305	**0.8485**

Table 2: F1 scores of three systems according to the sections in CTRs.

5.1 Baseline F1

In the test dataset, there are 500 samples without interventions on the statements, which are used as the control set. Table 2 shows the F1 scores of three systems according to the sections in the CTRs. The Eligibility section has the lowest F1 scores compared to the other sections in CTRs. Several factors have increased the difficulties for eligibility verification statements. First, the premises, i.e., criteria for inclusion and exclusion for clinical trials, are much longer than those for other sections. LLMs cannot always extract all relevant information from very long contexts. The verification of statements regarding eligibility often requires multi-step reasoning capabilities. The lack of domain-specific knowledge and common sense can also lead to verification errors. For the sections Intervention, Adverse Events, and Results, the major error type is numerical reasoning.

5.2 Consistency

Compared to the other two metrics, the consistency score of our self-consistent CoT system has the worst ranking. There are 4136 samples with semantic-preserving interventions, and they can be classified into five groups: paraphrase preserving, contradiction preserving, numerical paraphrase preserving, numerical contradiction preserving, and definitions preserving. Table 3 summarizes the consistency scores for each system according to

these categories. Compared to the other categories, self-consistent CoT and CoT greedy systems have worse performance regarding definitions and numerical paraphrase interventions. For the intervention regarding definitions, extra factual statements have been added to the original statements. A definition-intervention to the original statement in Figure 1, e.g., is:
Over 1/2 patients in the primary trial treated with Lapatinib 1000 mg + Nab-Paclitaxel experienced either a confirmed complete response (CR) or a confirmed partial response (PR). bladder solitary fibrous tumor is a solitary fibrous tumor that arises from the bladder. most tumors are benign.
Since the sentence *bladder solitary fibrous ...* is irrelevant to the clinical report, our self-consistent CoT tends to verify it as Contradiction (7 Contradictions : 3 Entailments in 10 generated reasoning chains).

For the category of numerical paraphrasing, there are 224 statements modified from 90 original statements. For the original set, we have around 81% accuracy (73 / 90), and for the modified set around 72% (162 / 224). We classify the errors, in total 34, where original statements are verified correctly and modified statements are verified wrong, into three groups: the conversion of fractional numbers, decimals, and percentages; the conversion of time, e.g., months to weeks, months to days, etc.; the conversion of units, e.g., mm to cm, mg to micrograms, etc. Among 34 misclassified statements, the system can generate at least one completely correct reasoning chain out of 10 reasoning chains for 17 statements. This indicates that the model has the knowledge for these conversions. However, our instruction-tuned model has difficulties utilizing this knowledge. The model also has difficulties understanding some paraphrased statements, e.g., from *over 1/2 patients* to *over 0.5 patients*, from *5% of patients* to *0.05 of patients*, etc. Another issue is some conversions are not quite precise, e.g., *9 months* to *36 weeks*, *6 months* to *180 days*. The model often contradicts the modified statements since they are not the same (9 months equals about 39 weeks).

5.3 Faithfulness

The contrast set for evaluating the faithfulness of the system is generated from 250 samples from the control set. All the 250 statements in the control set have the label Entailment. With contradiction intervention, 864 samples are created with the label Contradiction, 114 of which are numerically intervened. Table 4 summarizes the faithfulness scores of each system after contradiction intervention types. The scores show that both CoT greedy and self-consistent CoT have a significant performance improvement over label-only prediction. This underscores the critical role of the extra reasoning chain generation step for faithful verification.

Model	Consistency				
	para.	cont.	num. para.	num. cont.	defi.
Label-only Greedy	0.7807	0.7613	**0.7411**	0.9568	**0.6553**
CoT Greedy	0.7780	0.8427	0.7321	0.9568	0.5780
Self-consistent CoT	**0.8020**	**0.8440**	0.7232	**0.9877**	0.5720

Table 3: Consistency scores of three systems according to semantic-preserving intervention types

Model	Faithfulness	
	cont. alter.	num. cont. alter.
Label-only Greedy	0.8040	0.8509
CoT Greedy	0.8933	0.9211
Self-consistent CoT	**0.8987**	**0.9386**

Table 4: Faithfulness scores of three systems according to intervention (altering) types

5.4 Self-consistent CoT and CoT Greedy

According to Table 1, our self-consistent CoT has improved the performance of the CoT greedy system regarding all metrics, namely baseline F1 of 1.31 percentage points, consistency score of 0.75 percentage points, and faithfulness score of 0.69 percentage points. However, the improvement is insignificant compared to Wang et al. (2023). We show two examples in Appendix C. The first example shows that self-consistent CoT corrects the error in CoT greedy, and the second one shows that self-consistent CoT fails to correct the error. One possible reason for the insignificant improvement is the number of generated reasoning chains. Due to computational limitations, we have only generated 10.36 reasoning chains on average, which is much less than the 40 reasoning chains in Wang et al. (2023). Another important reason is the aggregating mechanism of majority voting. From the above-mentioned second example, we can see the late simple aggregation of the verification labels is not enough for the binary classification problem. A more finegrained verification and integration of intermediate reasoning steps of CoT is needed to tackle the inconsistency problem further.

6 Conclusion

In this paper, we have reported our self-consistent CoT system for the SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. For comparison, we also reported the label-only greedy and CoT greedy inference systems. To achieve a trustworthy inference system, we utilized the CoT reasoning paradigm, not only predicting the verification labels but also rationales. We tackled CoT's inconsistency problem with self-consistent CoT. Compared to the greedy CoT system, we have improved the inference performance by generating multiple reasoning chains and verifying with majority voting. However, the performance improvement is limited. For future work, a more fine-grained evaluation of the correctness of the reasoning steps in the CoT paradigm is promising for solving complex reasoning tasks.

7 Acknowledgments

This work was carried out with the support of the German Federal Ministry of Education and Research (BMBF) within the project "DeFaktS" (Grant 16KIS1524K). This work was also supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and et al. 2023. Gpt-4 technical report.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and et al. 2024. Mixtral of experts.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama AI research at SemEval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003, Toronto, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. *ArXiv*, abs/1804.08838.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and et al. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. 2023. Natural language reasoning, a survey.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. Pytorch FSDP: experiences on scaling fully sharded data parallel. *Proc. VLDB Endow.*, 16(12):3848–3860.

A Prompts

A.1 Instruction prompt for GPT-4

Figure 3 shows an instruction example for GPT-4 to verify the statements in the train and validation datasets with CoT.

<s>[INST] **Primary clinical trial report**:
Adverse Events 1: Total: 12/32 (37.50%) Anaemia 0/32 (0.00%) Neutropenia 1/32 (3.13%) Thrombocytopenia 4/32 (12.50%) Atrial fibrillation 1/32 (3.13%) Cardiac failure congestive 1/32 (3.13%) Myocardial ischaemia 1/32 (3.13%) Abdominal discomfort 0/32 (0.00%) Ascites 1/32 (3.13%) Constipation 0/32 (0.00%) Rectal haemorrhage 1/32 (3.13%) Vomiting 1/32 (3.13%) Fatigue 1/32 (3.13%) Adverse Events 2: Total: 8/20 (40.00%) Anaemia 1/20 (5.00%) Neutropenia 0/20 (0.00%) Thrombocytopenia 1/20 (5.00%) Atrial fibrillation 0/20 (0.00%) Cardiac failure congestive 0/20 (0.00%) Myocardial ischaemia 0/20 (0.00%) Abdominal discomfort 1/20 (5.00%) Ascites 0/20 (0.00%) Constipation 2/20 (10.00%) Rectal haemorrhage 0/20 (0.00%) Vomiting 0/20 (0.00%) Fatigue 0/20 (0.00%)
Statement: Most of the cases of CHF in the primary trial, were in cohort 1.
Verify whether the statement is entailed in the primary clinical trial report with Entailment or Contradiction. [/INST]
Let's verify it step by step:

Figure 3: Example prompt for GPT-4

A.2 Example for CoT instruction-tuning dataset

Figure 4 shows an example of CoT instruction-tuning dataset. The rationales in CoT are generated by GPT-4 and we add the final verification label to CoT for convenience by extracting the labels.

A.3 Example for label-only instruction-tuning dataset

Figure 5 shows an example of the label-only instruction-tuning dataset, where only the gold label is appended to the (premise, statement) pair without CoT rationales.

B Implementation Details

B.1 Instruction-Tuning

For the instruction-tuning, we have applied PyTorch Fully Sharded Data Parallel (FSDP) from Meta AI (Zhao et al., 2023). The training script is adapted based on llama-recipes[3]. We train Mixtral-8x7B-Instruct on a cluster with 16 A100-40GB GPUs with a batch size of 1 and a learning rate of 0.0001. Mixed precision and gradient checkpointing have been applied to save memory. Evaluated on the validation dataset, the model achieves the best performance with 5 epochs. Following Kanakarajan and Sankarasubbu (2023), we include the validation set as part of the training data for the final submission.

<s>[INST] **Primary clinical trial report**:
Adverse Events 1: Total: 12/32 (37.50%) Anaemia 0/32 (0.00%) Neutropenia 1/32 (3.13%) Thrombocytopenia 4/32 (12.50%) Atrial fibrillation 1/32 (3.13%) Cardiac failure congestive 1/32 (3.13%) Myocardial ischaemia 1/32 (3.13%) Abdominal discomfort 0/32 (0.00%) Ascites 1/32 (3.13%) Constipation 0/32 (0.00%) Rectal haemorrhage 1/32 (3.13%) Vomiting 1/32 (3.13%) Fatigue 1/32 (3.13%) Adverse Events 2: Total: 8/20 (40.00%) Anaemia 1/20 (5.00%) Neutropenia 0/20 (0.00%) Thrombocytopenia 1/20 (5.00%) Atrial fibrillation 0/20 (0.00%) Cardiac failure congestive 0/20 (0.00%) Myocardial ischaemia 0/20 (0.00%) Abdominal discomfort 1/20 (5.00%) Ascites 0/20 (0.00%) Constipation 2/20 (10.00%) Rectal haemorrhage 0/20 (0.00%) Vomiting 0/20 (0.00%) Fatigue 0/20 (0.00%)
Statement: Most of the cases of CHF in the primary trial, were in cohort 1.
Verify whether the statement is entailed in the primary clinical trial report with Entailment or Contradiction. [/INST]
Let's verify it step by step: The statement refers to cases of Cardiac Heart Failure (CHF) in the primary trial. Looking at the data from the primary clinical trial report, we can see that in Adverse Events 1 (cohort 1), there was 1 case of Cardiac failure congestive out of 32 participants, which is 3.13%. In Adverse Events 2 (cohort 2), there were 0 cases of Cardiac failure congestive out of 20 participants, which is 0.00%.
Therefore, the statement "Most of the cases of CHF in the primary trial, were in cohort 1" is Entailed in the primary clinical trial report.
Verification: Entailment </s>

Figure 4: An example of CoT instruction-tuning dataset

B.2 Model Inference

We use the transformers[4] library to generate responses with instruction-tuned models. There are two setups for generation, greedy decoding and temperature sampling. Under temperature sampling, we set $T = 0.7$ and $k = 50$ in top-k. For each pair, we set `num_return_sequence` as 10. Occasionally, there are duplicates in the returned sequences. If the result is tied with majority voting, we further generate reasoning chains with different seeds. The maximum number of generated chains for one pair is 25. Table 5 shows the distribution for numbers of generated distinct reasoning chains in the test dataset. On average, we generate 10.36 reasoning chains for each pair in the test dataset.

[3] https://github.com/facebookresearch/llama-recipes/tree/main

[4] https://github.com/huggingface/transformers

<s>[INST] **Primary clinical trial report**:
Adverse Events 1: Total: 12/32 (37.50%) Anaemia 0/32 (0.00%) Neutropenia 1/32 (3.13%) Thrombocytopenia 4/32 (12.50%) Atrial fibrillation 1/32 (3.13%) Cardiac failure congestive 1/32 (3.13%) Myocardial ischaemia 1/32 (3.13%) Abdominal discomfort 0/32 (0.00%) Ascites 1/32 (3.13%) Constipation 0/32 (0.00%) Rectal haemorrhage 1/32 (3.13%) Vomiting 1/32 (3.13%) Fatigue 1/32 (3.13%) Adverse Events 2: Total: 8/20 (40.00%) Anaemia 1/20 (5.00%) Neutropenia 0/20 (0.00%) Thrombocytopenia 1/20 (5.00%) Atrial fibrillation 0/20 (0.00%) Cardiac failure congestive 0/20 (0.00%) Myocardial ischaemia 0/20 (0.00%) Abdominal discomfort 1/20 (5.00%) Ascites 0/20 (0.00%) Constipation 2/20 (10.00%) Rectal haemorrhage 0/20 (0.00%) Vomiting 0/20 (0.00%) Fatigue 0/20 (0.00%)
Statement: Most of the cases of CHF in the primary trial, were in cohort 1.
Verify whether the statement is entailed in the primary clinical trial report with Entailment or Contradiction. [/INST]
Verification: Entailment </s>

Figure 5: An example of label-only instruction-tuning dataset

Number of Chains	Number of Samples
10	5150
9	104
15	101
20	90
25	36
19	8
17	5
16	2
8, 14, 18, 24	1

Table 5: Distribution of numbers of generated distinct reasoning chains for test dataset

C Reasoning Chains Generation

Figure 6 shows an example where CoT greedy has wrong verification while self-consistent CoT has correct verification with majority voting. Figure 7 shows an example where both CoT greedy and self-consistent CoT have wrong verification.

<s>[INST] **Primary clinical trial report**: INTERVENTION 1: Arm 1 Herceptin/navelbine INTERVENTION 2: Arm 2 Taxotere/carboplatin/herceptin
Statement: only one arm of the primary clinical trial makes use of trastuzumab as part of its intervention
Verify whether the statement is entailed in the primary clinical trial report with Entailment or Contradiction. [/INST]
Let's verify it step by step:

Gold Label: Contradiction

CoT Greedy:
The statement mentions that only one arm of the primary clinical trial makes use of trastuzumab as part of its intervention. In the clinical trial report, Herceptin is mentioned in both Intervention 1 and Intervention 2. Herceptin is a brand name for trastuzumab, a monoclonal antibody used to treat certain types of breast cancer. Therefore, the statement is Entailed in the primary clinical trial report.
Verification: Entailment

Self-consistent CoT
Generation 1:
The statement mentions that only one arm of the primary clinical trial makes use of trastuzumab as part of its intervention. Trastuzumab is a monoclonal antibody used to treat certain types of breast cancer. It is also known by its brand name, Herceptin. In the primary clinical trial report, Herceptin is mentioned in both Intervention 1 (Arm 1) and Intervention 2 (Arm 2). Therefore, both arms of the trial are using trastuzumab (Herceptin) in their interventions. So, the statement is a Contradiction to the information provided in the primary clinical trial report.
Verification: Contradiction
Generation 2:
The statement mentions that only one arm of the primary clinical trial makes use of trastuzumab as part of its intervention. Looking at the clinical trial report, we can see that Herceptin is used in both arms of the trial. Herceptin is a brand name for trastuzumab, a medication used to treat certain types of breast cancer. Therefore, the statement is a Contradiction because both arms of the trial use trastuzumab, not just one.
Verification: Contradiction
Generation 3:
The statement mentions that only one arm of the primary clinical trial makes use of trastuzumab as part of its intervention. Trastuzumab is the generic name for Herceptin. Looking at the clinical trial report, we see that Herceptin is indeed used in both interventions. In Intervention 1, Herceptin is used in combination with navelbine. In Intervention 2, Herceptin is used in combination with Taxotere and carboplatin. Therefore, the statement contradicts the information given in the primary clinical trial report. The correct answer is Contradiction.
Verification: Contradiction
Generation 4:
The statement mentions that only one arm of the primary clinical trial makes use of trastuzumab as part of its intervention. In the clinical trial report, Herceptin is mentioned in both Intervention 1 and Intervention 2. Herceptin is a brand name for trastuzumab, a monoclonal antibody used to treat certain types of breast cancer. Therefore, the statement is Entailed in the primary clinical trial report.
Verification: Entailment
Generation 5:
The clinical trial report mentions two interventions. In Intervention 1 (Arm 1), Herceptin is mentioned. Herceptin is a brand name for the drug trastuzumab. In Intervention 2 (Arm 2), there is no mention of Herceptin or trastuzumab. Therefore, the statement "only one arm of the primary clinical trial makes use of trastuzumab as part of its intervention" is Entailed in the primary clinical trial report. Verification: Entailment
.....

Figure 6: An example of self-consistent CoT successfully corrects the error in CoT greedy verification. We only show 5 generations of self-consistent CoT. The ratio of 10 self-consistent CoT generations is 8 Contradictions: 2 Entailments.

<s>[INST] **Primary clinical trial report**: Adverse Events 1: Total: 10/30 (33.33%) Hemoglobin decreased 2/30 (6.67%) Abdominal pain 1/30 (3.33%) Colitis 1/30 (3.33%) Diarrhea 7/30 (23.33%) Nausea 2/30 (6.67%) Rectal hemorrhage 1/30 (3.33%) Fatigue 1/30 (3.33%) Skin infection 1/30 (3.33%) Neutrophil count decreased 1/30 (3.33%) Platelet count decreased 3/30 (10.00%) Dehydration 1/30 (3.33%)
Secondary clinical trial report: Adverse Events 1: Total: 3/6 (50.00%) Anaemia 0/6 (0.00%) Febrile neutropenia 0/6 (0.00%) Neutropenia 0/6 (0.00%) Bradycardia 0/6 (0.00%) Diarrhoea 0/6 (0.00%) Pancreatitis 0/6 (0.00%) Vomiting 0/6 (0.00%) Disease progression 0/6 (0.00%) Fatigue 0/6 (0.00%) Pyrexia 0/6 (0.00%) Cholelithiasis 0/6 (0.00%) Hepatic pain 0/6 (0.00%) Bacteraemia 0/6 (0.00%) Adverse Events 2: Total: 3/6 (50.00%) Anaemia 0/6 (0.00%) Febrile neutropenia 0/6 (0.00%) Neutropenia 1/6 (16.67%) Bradycardia 0/6 (0.00%) Diarrhoea 0/6 (0.00%) Pancreatitis 0/6 (0.00%) Vomiting 0/6 (0.00%) Disease progression 0/6 (0.00%) Fatigue 1/6 (16.67%) Pyrexia 1/6 (16.67%) Cholelithiasis 0/6 (0.00%) Hepatic pain 1/6 (16.67%) Bacteraemia 0/6 (0.00%)
Statement: diarrhoea was uncommon among the primary trial participants compared to the secondary trial participants
Verify whether the statement is entailed in the clinical trial reports with Entailment or Contradiction. [/INST]
Let's verify it step by step:

Gold Label: Contradiction

CoT Greedy:
In the primary clinical trial report, the occurrence of diarrhea is 7/30 (23.33%). In the secondary clinical trial report, the occurrence of diarrhea is 0/6 (0.00%) in both Adverse Events 1 and Adverse Events 2. Therefore, the statement "diarrhoea was uncommon among the primary trial participants compared to the secondary trial participants" is Entailed in the clinical trial reports.
Verification: Entailment

Self-consistent CoT
Generation 1:
The statement claims that diarrhoea was uncommon among the primary trial participants compared to the secondary trial participants. Looking at the primary clinical trial report, it shows that the incidence of diarrhoea was 7 out of 30 participants, which is approximately 23.33%. In the secondary clinical trial report, the incidence of diarrhoea is 0 out of 6 participants in both Adverse Events 1 and Adverse Events 2, which is approximately 0.00%. Comparing these two reports, it is clear that diarrhoea was indeed more common in the primary trial participants than in the secondary trial participants. Therefore, the statement is a Contradiction to the information provided in the clinical trial reports.
Verification: Contradiction
Generation 2:
The statement says that diarrhoea was uncommon among the primary trial participants compared to the secondary trial participants. Looking at the primary clinical trial report, Diarrhea was reported in 7 out of 30 participants, which is approximately 23.33%. In the secondary clinical trial report, Diarrhoea was reported in 0 out of 6 participants in both Adverse Events 1 and Adverse Events 2, which is 0%. Comparing these two, it is clear that Diarrhoea was indeed more common in the primary trial participants than in the secondary trial participants. Therefore, the statement is a Contradiction to the information provided in the clinical trial reports.
Verification: Contradiction
Generation 3:
In the primary clinical trial report, the incidence of diarrhea was 7/30 (23.33%). In the secondary clinical trial report, the incidence of diarrhoea was 0/6 (0.00%) for both Adverse Events 1 and Adverse Events 2. Therefore, the statement "diarrhoea was uncommon among the primary trial participants compared to the secondary trial participants" is Entailed in the clinical trial reports.
Verification: Entailment
Generation 4:
In the primary clinical trial report, the occurrence of diarrhea is reported as 7/30 (23.33%). In the secondary clinical trial report, the occurrence of diarrhea is reported as 0/6 (0.00%) in both Adverse Events 1 and Adverse Events 2. Therefore, the statement "diarrhoea was uncommon among the primary trial participants compared to the secondary trial participants" is Entailed in the clinical trial reports.
Verification: Entailment
Generation 5:
In the primary clinical trial report, Diarrhea is reported in 7 out of 30 participants, which is approximately 23.33%. In the secondary clinical trial report, Diarrhoea is reported in 0 out of 6 participants in both Adverse Events 1 and Adverse Events 2, which is 0.00%. Therefore, the statement "diarrhoea was uncommon among the primary trial participants compared to the secondary trial participants" is Entailed in the clinical trial reports.
Verification: Entailment
.....

Figure 7: An example of self-consistent CoT fails to correct the error in CoT greedy verification. We only show 5 generations of self-consistent CoT. The ratio of 10 self-consistent CoT generations is 7 Entailments: 3 Contradictions.

Lisbon Computational Linguists at SemEval-2024 Task 2: Using A Mistral-7B Model and Data Augmentation

Artur Guimarães
INESC-ID and IST
University of Lisbon
Lisbon, Portugal
artur.guimas@gmail.com

Bruno Martins
INESC-ID and IST
University of Lisbon
Lisbon, Portugal
bruno.g.martins@tecnico.ulisboa.pt

João Magalhães
NOVA-LINCS
NOVA University of Lisbon
Lisbon, Portugal
jmag@fct.unl.pt

Abstract

This paper describes our approach to the SemEval-2024 safe biomedical Natural Language Inference for Clinical Trials (NLI4CT) task, which concerns classifying statements about Clinical Trial Reports (CTRs). We explored the capabilities of Mistral-7B, a generalist open-source Large Language Model (LLM). We developed a prompt for the NLI4CT task, and fine-tuned a quantized version of the model using an augmented version of the training dataset. The experimental results show that this approach can produce notable results in terms of the macro F1-score, while having limitations in terms of faithfulness and consistency. All the developed code is publicly available on a GitHub repository[1].

1 Introduction

Large Language Models (LLMs) currently achieve state-of-the-art performance on different Natural Language Processing (NLP) tasks, including in the assessment of textual entailment relations. However, these models are heavily susceptible to shortcut learning (Du et al., 2023), factual inconsistency (Xie et al., 2023), and performance degradation when exposed to data from specialized domains, such as in the case of medical data.

Noting the aforementioned challenges, Task 2 at SemEval-2024 addressed a safe biomedical Natural Language Inference for Clinical Trials (NLI4CT) task (Jullien et al., 2024), which concerns classifying statements about Clinical Trial Reports (CTRs). NLI4CT investigated the accuracy, faithfulness, and consistency of the reasoning performed by LLMs in this particular medical task. The goal of the task is to determine whether there is an entailment relation or a contradiction relation between CTRs and statements, making some type of claim about a single CTR or a pair of CTRs. Given the specific focus on assessing model faithfulness and consistency (i.e., the ability to make correct predictions for the correct reasons), the dataset associated to the task involved the systematic application of controlled interventions, either preserving or inverting the entailment relations originally generated by clinical domain experts. This way, the task investigated the robustness of NLI models in their representation of the semantic phenomena necessary for complex inference in clinical settings.

Our approach to the NLI4CT task involved the use of open-source LLMs, with good results in general purpose benchmarks[2] and capable of following task instructions. We opted for `Mistral-7B-Instruct-v0.2`[3] (Jiang et al., 2023), quantizing the model to 4-bits and simultaneously using Low-Rank Adaptation (LoRA) (Hu et al., 2021; Dettmers et al., 2023) to fine-tune the model to the NLI4CT task, using a slightly augmented version of the training dataset that features a mixture of manually curated and synthetic statements.

Our overall best submission to the task achieved a **macro F1-score** of 0.80 (1st place on the leaderboard), a **consistency** score of 0.72 (15th), and a **faithfulness** score of 0.83 (11th). Our method excels in classification accuracy, but fails at being robust to perturbations on the statements, i.e. predicting the same label on contradictory examples and different labels on paraphrased examples.

2 Background

The NLI4CT task concerns inferring if statements can be entailed by a given textual context, with each statement referring to one or two CTRs. These CTRs belong to a corpus consisting of 1000 different trials concerning breast cancer, extracted from

[1] https://github.com/araag2/SemEval2024-Task2

[2] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

[3] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1280–1287

Set	# Samples	Single - Compari.	Entail. - Contr.
Training	1700	60.9% - 39.1%	50% - 50%
Development	200	70% - 30%	50% - 50%
Pratice-test	2142	71.2% - 28.8%	34.1% - 65.9%
Test	5500	46.4% - 53.6%	33.5% - 66.5%

Table 1: The NLI4CT task dataset.

Set	# Interventions	Preserving - Altering (Label)
Pratice-test	1942 (90.7%)	82.7% - 27.3%
Test	5000 (90.9%)	82.7% - 27.3%

Table 2: Interventions over statements on the test splits.

the United States National Library of Medicine[4]. These trial reports are exclusively written in the English language and average 817 words in length.

CTRs are divided into four sections: **Eligibility Criteria**, describing a set of conditions to allow or exclude patients in the trial; **Interventions**, detailing all information about the conducted treatments; **Results**, outlining outcomes and experimental results gathered through the trial; and **Adverse Events**, reporting patient observations concerning symptoms and physiological signs. An instance of the NLI4CT dataset contains either one or two CTRs (i.e., cases denoted as single or comparison, respectively), a statement, a section marker, and a ground-truth label (i.e., entailment or contradiction). An example is shown next.

Listing 1 An instance from the NLI4CT dataset.

`Primary Trial:`
INTERVENTION 1:
• Letrozole, Breast Enhancement, Safety.
• Single arm of healthy postmenopausal women to have two breast MRI (baseline and post-treatment). Letrozole of 12.5 mg/day is given for three successive days just prior to the second MRI.

`Secondary Trial:`
INTERVENTION 1:
• FFDM Mammography Exam - LIP Algorithm
• Screening or diagnostic Full Field Digital Mammography (FFDM) exam
INTERVENTION 2:
• FFDM Mammography Exam - SIP Algorithm.
• The same 130 raw data images were externally reprocessed with the Siemens processing algorithm.

`Section:` Intervention
`Statement:` The primary trial and the secondary trial both used MRI for their interventions.
`Label:` Entailment.

The dataset[5] provided by the task organizers considered training, development, practice-test, and test splits (the last two without ground-truth labels during the competition), with a general statistical characterization provided in Table 1.

The first two splits, i.e. training and development, are similar to those used in the SemEval-2023 edition of the task (Jullien et al., 2023b), based on the work by Jullien et al. (2023a). These are two balanced sets, with mostly unique CTR-statement associations (i.e., statements that are not rephrasing or contradicting other ones). On the other hand, this composition contrasts with the practice-test and test splits, that are both imbalanced and almost solely composed of statements featuring interventions (e.g., paraphrasing, contradicting, or appending text) over a small set of original statements (< 10%), as show in Table 2. This distribution favours systems that focus on robustly classifying a small set of samples.

3 System Overview

We now describe our general approach to the SemEval-2024 NLI4CT task.

3.1 Choice of LLM

When deciding on how to build our NLI4CT system, we started by testing the zero-shot and few-shot capabilities of several open-source LLMs, before settling on the use of `Mistral-7B-Instruct-v0.2`. In addition to achieving good zero-shot results, this model also allowed us to process arbitrarily long input texts, which in this task is particularly relevant, since some CTRs can exceed 3000 tokens in length.

3.2 Model Prompting

A great deal of attention is currently given to prompting techniques, as the successful use of an LLM can be severely impaired by suboptimal prompts, and also since instruction fine-tuning (Chowdhery et al., 2022; Chung et al., 2022) is dependant on the prompt quality. In order to address the task of choosing a good prompt, we started by creating a prompt template that we deemed as suitable for the task at hand, sub-dividing our prompt into distinct parts (pre-pended with "$") that can latter be replaced with different textual realizations. The overall structure is illustrated next.

[4] https://clinicaltrials.gov/
[5] https://github.com/ai-systems/Task-2-SemEval-2024/blob/main/README.md

Listing 2 Overall prompt structure.

```
$task_description
$ctr_description

Primary Trial:
$primary_evidence

Secondary Trial:
$secondary_evidence

$statement_description
$statement

$option_description
```

Four of the parts are **sample independent**: `$task_description` provides a general description for the natural language inference task between CTRs and statements; `$ctr_description` delineates the general contents of a CTR and its different sections; `$statement_description` conveys the nature of the $statement; and lastly `$option_description` outlines the answers we expect from the model (e.g., an answer of YES or NO, depending on whether the CTR supports the statement). Conversely, `$primary_evidence`, `$secondary_evidence`, and `$statement` are **sample dependent**, as these parts should be replaced by the primary CTR, the secondary CTR (if applicable), and the statement, respectively.

We created 5 base prompts (see Appendix A.1) for each of the 4 sample independent parts, yielding 625 possible combinations for the general template. We evaluated all the combinations on the development set, and chose the prompt that yielded the top **macro F1-score**, which is shown in Listing 3.

3.3 Generating Answers

With the aforementioned template, we used the Python HuggingFace Transformers library[6] to generate answers with `Mistral-7B-Instruct-v0.2`, using as parameters `do_sample=True`, `top_k=5`, and `max_new_tokens=30`. We opted not to constrain the generation process, instead looking for sets of words, associated to each label, in the sequence of generated tokens. The words "Yes", "yes", and "entailment" were used for the entailment class, while the words "No", "no" and "contradiction" were used for the contradiction class. Preference was given to the first token in the sequence that belongs to either of the sets, and if none were found we label the instance as entailment.

[6] https://huggingface.co/docs/transformers/en/index

Listing 3 The best performing prompt.

<s>[INST]The objective is to examine semantic entailment relationships between individual sections of Clinical Trial Reports (CTRs) and statements articulated by clinical domain experts. CTRs elaborate on the procedures and findings of clinical trials, scrutinizing the effectiveness and safety of novel treatments. Each trial involves cohorts or arms exposed to distinct treatments or exhibiting diverse baseline characteristics.
Comprehensive CTRs comprise four sections: (1) ELIGIBILITY CRITERIA delineating conditions for patient inclusion, (2) INTERVENTION particulars specifying type, dosage, frequency, and duration of treatments, (3) RESULTS summary encompassing participant statistics, outcome measures, units, and conclusions, and (4) ADVERSE EVENTS cataloging signs and symptoms observed. Statements posit claims regarding the information within these sections, either for a single CTR or in comparative analysis of two. To establish entailment, the statement's assertion should harmonize with clinical trial data, find substantiation in the CTR, and avoid contradiction with the provided descriptions.
The following descriptions correspond to the information in one of the Clinical Trial Report (CTR) sections.

Primary Trial:
$primary_evidence

Secondary Trial:
$secondary_evidence

Reflect upon the ensuing statement crafted by an expert in clinical trials.
$statement
Respond with either YES or NO to indicate whether it is possible to determine the statement's validity based on the Clinical Trial Report (CTR) information, with the statement being supported by the CTR data and not contradicting the provided descriptions.[/INST] **Answer:**

3.4 Data Augmentation

The NLI4CT dataset features 1700 training instances and 200 development instances, which is perhaps insufficient for fine-tuning an LLM in order to generalize to a testing split that is almost thrice as large. We decided to augment the available data, and created the 3 different training splits outlined in Table 3.

Set	# Samples	Single - Compari.	Entail. - Contr.
Train_Manual	2344	61.8% - 38.2%	50% - 50%
Train_Manual-Synthetic	3720	63.7% - 36.3%	50% - 50%
Train_Full-Synthetic	11011	60.9% - 39.1%	46.3% - 53.7%

Table 3: Results from task data augmentation.

The three new sets were constructed as follows:

- **Train_Manual:** Starting from the train split, we added queries created by using pre-existing samples with the entailment class, negating them using the Python negate li-

brary[7] (i.e., to generate corresponding contradiction examples), and also manually paraphrasing the original instance (i.e., to generate different entailment samples). All 644 additional samples that were generated through this procedure were manually curated;

- **Train_Manual-Synthetic:** starting from the **Train_Manual** dataset, we added 1376 new automatically generated instances to this set: half of the new instances were generated with the negate library, and the other half were generated by paraphrasing existing statements using the `Mistral-7B-Instruct-v0.2` model;

- **Train_Full-Synthetic:** Starting from the train split, we added 9311 new samples, using the negate library on entailment instances, and the `Mistral-7B-Instruct-v0.2` model to paraphrase each original statement 5 times.

3.5 Instruction Fine-tuning

Noting that `Mistral-7B-Instruct-v0.2` is a generalist instruction fine-tuned model, we sought to fine-tune this LLM to the NLI4CT task, using the aforementioned instructions. To improve the training efficiency and support very long sequences (i.e., up to 6000 tokens), we quantized the model to 4-bit representations of the parameters, and used LoRA (Hu et al., 2021). Model training used a supervised fine-tuning objective based on auto-regressive language modelling, completing the input instruction with the correct label for each instance (i.e., outputting either "Yes" or "No" after "Answer:" in the prompt). The implementation relied on the PEFT[8] and TRL[9] Python libraries.

4 Experimental Setup

Making official submissions to the task leaderboard required the participants to submit full runs of the test set, outputting a label for each of its instances. We obtained the labels for each instance by following the procedure described in Subsection 3.3.

The task uses the following evaluation measures: **macro F1-score**, i.e. the arithmetic mean of precision and recall, averaged over the two classes; **Faithfulness**, i.e. a measure created to assess the capacity of model to arrive at the correct prediction for the correct reason, calculated by measuring the ability of model to change its prediction label after semantically altering a statement; and **Consistency**, which completes faithfulness by measuring the ability of a model in outputting the same prediction for semantically equivalent statements[10]. We evaluated our runs using the official metrics obtained from the leaderboard.

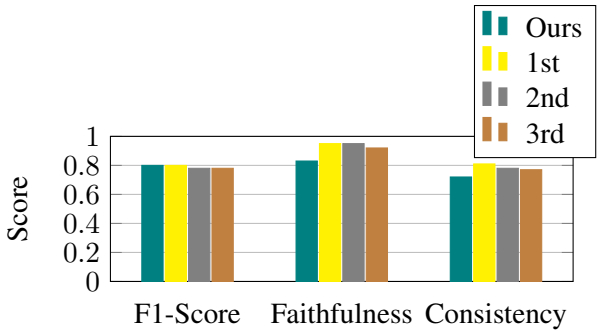


Figure 1: Comparison of top submissions against our system, according to different evaluation metrics.

Following the training procedure described in Section 3.5, we tested different combinations of training data (as described in Section 3.4). The full set of hyper-parameters associated to our best run can be found in Appendix A.2. All the different runs used Python libraries and packages that can be found in our GitHub repository[11].

5 Results and Discussion

Table 4 presents our most important results, showing the best result that we achieved with each training set. In turn, Figure 1 compares our overall best run with the top three submissions, per metric.

Trained Sets	F1-Score	Faithfulness	Consistency
None (Zero-Shot)	0.67 (3)	0.61 (8)	0.53 (8)
Train	0.81 (2)	0.72 (3)	0.69 (2)
Train_Manual	**0.82 (9)**	0.76 (9)	0.71 (9)
Train_Manual-Synthetic	0.80 (1)	**0.83 (1)**	**0.72 (2)**
Train_Full-Synthetic	0.78 (1)	0.78 (0)	0.71 (0)

Table 4: Results on different training datasets.

We tested the LLM without any training (i.e., zero-shot results), and fine-tuning with the base and augment datasets, all with our best instruction format. As expected, there is a significant difference in performance towards fine-tuned models. Overall, a mixture between manually curated sam-

[7] https://github.com/dmlls/negate
[8] https://huggingface.co/docs/peft/en/index
[9] https://huggingface.co/docs/trl/en/index
[10] https://github.com/ai-systems/Task-2-SemEval-2024/blob/main/evaluate.py
[11] https://github.com/araag2/SemEval2024-Task2/blob/main/environment.yml

ples and synthetically generated ones performed best (Train_Manual-Synthetic, as described on Section 4), outperforming the best run that did not use any data augmentation. If more instances could have been manually curated, specifically targeting adversarial re-writes of the same statements, we hypothesize that results could be improved further. Even though **Train_Full-Synthetic** corresponds to the largest training set (i.e., featuring 11011 samples), the lack of quality in the automatically generated statements potentially impaired the **F1-score** while also limiting **consistency** and **faithfulness**.

The run trained with **Train_Manual-Synthetic** corresponds to our best overall result. When compared to the top submissions, we can see that our F1-score corresponds to a tie with another system in the 1st place of the leaderboard. However, results are much worse in the other two metrics, with significant differences between the top systems (i.e., with scores of 0.95 in faithfulness and 0.81 in consistency) and our submission.

In the post-task phase of the competition, ground-truth labels for all examples were released, specifying which type of interventions were made in each instance. Therefore, we are now able to analyse our system's errors (see Table 5), to support a discussion on the main short-comings of our work.

Type of Error	# Occurrences / # Total Samples
Base Statement Errors	99 / 500 (19.8%)
Intervention Errors	1328 / 5000 (26.7%)
Total Errors	1427 / 5500 (25.9%)
Label Preserving Intervention Errors	1177 / 1328 (88.6%)
Label Altering Intervention Errors	151 / 1328 (11.4%)
Paraphrasing Errors	344 / 1500 (22.9%)
Text Appending Errors	609 / 1500 (40.6%)
Contradicting Errors	293 / 1500 (19.5%)
Numerical Paraphrasing Errors	58 / 224 (25.9%)
Numerical Contradicting Errors	24 / 276 (8.7%)

Table 5: Error analysis for our best overall run, categorizing errors by intervention types.

Comparing all errors across the different instance types, the average error rate is much higher on intervention errors (26.7%) against base statement errors (19.8%), which is to be expected as our training sets had fewer examples of this type. Specifically, we can see that label preserving interventions (88.6%) have a high percentage of errors. Our system can identify instances which suffered contradictory interventions with an error rate of 19.5% for textual changes, and 8.7% for numerical changes. Instances that were perturbed with paraphrasing cause an error rate of 22.9%, while numerical paraphrasing errors correspond to 25.9%. At the worst end we have the samples with text appended to the end, which causes an error rate of 40.6%. Note that we did did not explicitly augment the training instances by appending text to the existing statements, and the absence of examples like this was very costly in terms of the final results.

6 Conclusions

Adapting evaluation methodologies to better inform the safe deployment of LLMs in critical domains is an urgent necessity. The NLI4CT task at SemEval-2024 addressed this specific concern, and through our participation we improved our understanding on how LLMs can be fine-tuned to encompass robust results on clinical natural language inference. Overall, our results show that the simple fine-tuning of an open-source LLM to this specific task can achieve notable results in terms of the macro-averaged F1, although with limitations in terms of faithfulness and consistency. Augmenting the data with high-quality curated examples can improve result quality, although augmenting the training set with synthetic examples requires careful quality control.

For future work we would like to explore the following ideas:

- Test our general approach with different models, specifically considering models fine-tuned in the medical domain (e.g., models like `qCammel-70-x`[12] or `BioMistral`[13]);
- Refining the considered prompt through recently-proposed prompt optimization methods (Wen et al., 2023; Guo et al., 2023), instead of relying on manually curated prompts;
- Incorporating additional training data, e.g. by generating a more diverse set of instances from the CTR data made available in the context of other shared tasks (e.g., the CTR data from the Text Retrieval Conference (TREC) clinical trials track[14]);
- Carefully curating a new training set, with a focus on statement interventions rather than quantity of base statements, in order to better guide the model into understanding the nuances of textual and numerical paraphrasing/contradiction.

[12] https://huggingface.co/augtoma/qCammel-70-x
[13] https://huggingface.co/BioMistral
[14] https://www.trec-cds.org/

Acknowledgments

This research was supported by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (i.e., the Center For Responsible AI), and also by Fundação para a Ciência e Tecnologia (FCT), through the project with reference UIDB/50021/2020 (DOI:10.54499/UIDB/50021/2020).

References

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv:2210.11416*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized llms. *arXiv:2305.14314*.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv:2309.08532*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv:2106.09685*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv:2310.06825*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *arXiv:2302.03668*.

Qianqian Xie, Edward J Schenck, He S Yang, Yong Chen, Yifan Peng, and Fei Wang. 2023. Faithful ai in medicine: A systematic review with large language models and beyond. *medRxiv*.

A Appendix

We now present additional details about the prompt considered for instructing the `Mistral-7B-Instruct-v0.2` model, and about the hyper-parameters considered for model fine-tuning.

A.1 Base Descriptions For Each Prompt Part

This section presents the five different alternatives that were considered for the different parts of the `Mistral-7B-Instruct-v0.2` prompt.

A.1.1 Task Description Part

1 : Consider the task of determining semantic entailment relations between individual sections of Clinical Trial Reports (CTRs) and statements made by clinical domain experts. Note that CTRs outline the methodology and findings of a clinical trial, which are conducted to assess the effectiveness and safety of new treatments. Each trial involves 1-2 patient groups, called cohorts or arms, and these groups may receive different treatments, or have different baseline characteristics. The complete CTRs contain 4 sections, corresponding to (1) a list of the ELIGIBILITY CRITERIA corresponding to the conditions for patients to be allowed to take part in the clinical trial, (2) a description for the INTERVENTION that specifies the type, dosage, frequency, and duration of treatments being studied, (3) a summary of the RESULTS, detailing aspects such as the number of participants in the trial, the outcome measures, the units, and the conclusions, and (4) a list of ADVERSE EVENTS corresponding to signs and symptoms observed in patients during the clinical trial. In turn, the statements are sentences that make some type of claim about the information contained in one of the aforementioned sections, either considering a single CTR or comparing two CTRs. In order for the entailment relationship to be established, the claim in the statement should be related to the clinical trial information, it should be supported by the CTR, and it must not contradict the provided descriptions.

2 : You are tasked with determining support relationships between individual sections of Clinical Trial Reports (CTRs) and clinical statements. CTRs detail the methodology and findings of clinical trials, assessing effectiveness and safety of new treatments. CTRs consist of 4 sections: (1) ELIGIBILITY CRITERIA listing conditions for patient participation, (2) INTERVENTION description specifying type, dosage, frequency, and duration of treatments, (3) RESULTS summary detailing participants, outcome measures, units, and conclusions, and (4) ADVERSE EVENTS listing signs and symptoms observed. Statements make claims about information in these sections, either for a single CTR or comparing two.

3 : Evaluate the semantic entailment between individual sections of Clinical Trial Reports (CTRs) and statements issued by clinical domain experts. CTRs expound on the methodology and outcomes of clinical trials, appraising the efficacy and safety of new treatments. The statements, on the other hand, assert claims about the information within specific sections of CTRs, for a single CTR or comparative analysis of two. For entailment validation, the statement's claim should align with clinical trial information, find support in the CTR, and refrain from contradicting provided descriptions.

4 : The objective is to examine semantic entailment relationships between individual sections of Clinical Trial Reports (CTRs) and statements articulated by clinical domain experts. CTRs elaborate on the procedures and findings of clinical trials, scrutinizing the effectiveness and safety of novel treatments. Each trial involves cohorts or arms exposed to distinct treatments or exhibiting diverse baseline characteristics. Comprehensive CTRs comprise four sections: (1) ELIGIBILITY CRITERIA delineating conditions for patient inclusion, (2) INTERVENTION particulars specifying type, dosage, frequency, and duration of treatments, (3) RESULTS summary encompassing participant statistics, outcome measures, units, and conclusions, and (4) ADVERSE EVENTS cataloging signs and symptoms observed. Statements posit claims regarding the information within these sections, either for a single CTR or in comparative analysis of two. To establish entailment, the statement's assertion should harmonize with clinical trial data, find substantiation in the CTR, and avoid contradiction with the provided descriptions.

5 : Consider the problem of assessing semantic entailment connections between distinct sections of Clinical Trial Reports (CTRs) and statements put forth by clinical domain experts. To establish entailment, the statement's assertion should be supported from the CTR, not contradicting the provided descriptions. In brief, CTRs elucidate the procedures and findings of clinical trials, evaluating the efficacy and safety of emerging treatments. Complete CTRs encompass four sections: (1) ELIGIBILITY CRITERIA specifying conditions for patient inclusion, (2) INTERVENTION details on the type, dosage, frequency, and duration of treatments, (3) RESULTS summarizing the participant statistics, outcome measures, units, and conclusions, and (4) ADVERSE EVENTS listing observed signs and symptoms. Statements advance claims about the information within these sections, either for a single CTR or in a comparative analysis of two CTRs.

A.1.2 CTR Description Part

1 : The following descriptions correspond to the information in one of the Clinical Trial Report (CTR) sections.

2 : The provided descriptions coincide with the content in a specific section of Clinical Trial Reports (CTRs), detailing relevant information to the trial.

3 : The provided descriptions correspond to the content found in one of the four standard clinical trial report sections.

4 : The provided descriptions pertain to the contents found within one of the sections of Clinical Trial Reports (CTRs).

5 : The descriptions that follow correspond to the information contained in one of the standard sections of the clinical trial reports.

A.1.3 Statement Description Part

1 : Consider also the following statement generated by a clinical domain expert, a clinical trial organizer, or a medical researcher.

2 : Contemplate the ensuing statement formulated by a clinical expert or researcher.

3 : Review the subsequent statement provided by an expert in clinical trials, attending to the medical terminology and carefully addressing any ambiguities.

4 : Deliberate upon the subsequent statement formulated by an healthcare practitioner, a coordinator of clinical trials, or a medical researcher.

5 : Reflect upon the ensuing statement crafted by an expert in clinical trials.

A.1.4 Option Description Part

1 : Answer YES or NO to the question of whether one can conclude the validity of the statement with basis on the clinical trial report information.

2 : Indicate with either YES or NO whether it is possible to determine the validity of the statement based on the Clinical Trial Report (CTR) descriptions. An answer of YES means that the statement is supported by the CTR descriptions, not contradicting the provided information.

3 : Provide a YES or NO response indicating if it's possible to assess the statement's validity based on the information presented in the clinical trial report descriptions. Do this by interpreting the medical terminology and the context in both the report and the statement, carefully addressing any ambiguities or gaps in the provided information.

4 : Respond with either YES or NO to indicate whether it is possible to determine the statement's validity based on the Clinical Trial Report (CTR) information, with the statement being supported by the CTR data and not contradicting the provided descriptions.

5 : Indicate with a YES or NO response whether it is possible to assess the statement's validity based on the clinical trial report data.

A.2 Full List of Hyper-Parameters

The full list of hyper-parameters considered for model fine-tuning can be seen in the source-code in our GitHub repository[15].

The chosen parameters concerning model quantization options are as follows.

```
load_in_4bit = True
bnb_4bit_quant_type = "nf4"
bnb_4bit_compute_dtype = torch.bfloat16
bnb_4bit_use_double_quant = False
```

The parameters concerning the use of Low-Rank Adaptation (LoRA) are as follows.

[15] https://github.com/araag2/SemEval2024-Task2/blob/main/finetune_Mistral.py

```
lora_r = 64
lora_dropout = 0.1
lora_alpha = 16
bias = "none"
```

Finally, the general model training hyper-parameters are as follows.

```
train_epochs = 5
batch_size = 2
gradient_accumulation_steps = 4
learning_rate = 2e-5
pooling = "mean"
max_sequence_length = 6000
```

GIL-IIMAS UNAM at SemEval-2024 Task 1: SAND: An In Depth Analysis of Semantic Relatedness Using Regression and Similarity Characteristics.

F. López-Ponce[1], **Ángel Cadena**[1,2], **K. Salas-Jimenez**[1,2],
D. Preciado Márquez[1], **G. Bel-Enguix**[1,3]
[1]Grupo de Ingeniería Lingüística - UNAM
[2]Posgrado en Ciencias e Ingeniería de la Computación - UNAM
[3]Departament de Filologia Catalana i Lingüística General - Universitat de Barcelona
{francisco.lopez.ponce,karla_dsj}@ciencias.unam.mx
{angelcaden,davidpreciado115}@gmail.com
gbele@iingen.unam.mx

Abstract

The STR shared task aims at detecting the degree of semantic relatedness between sentence pairs in multiple languages. Semantic relatedness relies on elements such as topic similarity, point of view agreement, entailment, and even human intuition, making it a broader field than sentence similarity. The **GIL-IIMAS UNAM** team proposes a model based in the SAND characteristics composition (Sentence Transformers, AnglE Embeddings, N-grams, Sentence Length Difference coefficient) and classical regression algorithms. This model achieves a 0.83 Spearman Correlation score in the English test, and a 0.73 in the Spanish counterpart, finishing just above the SemEval baseline in English, and second place in Spanish.

1 Introduction

The Semantic Textual Relatedness (STR) task (Ousidhoum et al., 2024b) aims at creating systems that measure STR on pairs of sentences based on their closeness in meaning (Abdalla et al., 2023). This task is comprised of three tracks. Tracks A and B focus on monolingual models. Track A only accepts supervised models trained with the available tagged datasets, whereas track B focuses on the unsupervised approach relying on the same datasets but without the tagged score. Track C is the cross-lingual case where the target language has to follow an unsupervised approach. The datasets provided consist of sentence pairs that were sampled from various semantic similarity datasets.

This task expands upon classic sentence similarity comparisons, encouraging the use models and algorithms capable of analyzing more than meaning of a pair of sentences, focusing deeper on characteristics such as the syntactic structure of the sentences, lexicon relationships, as well as meaning and emotion. The GIL-IIMAS UNAM team participated in Track A. Although it included nine languages we have only worked with the Spanish and English dataset.

Track A is a regression problem since each dataset contains the sentence pairs as well as a corresponding sentence relatedness score that ranges from 0 to 1. The evaluation compares the ground values in the test set with the proposed model's prediction, meaning track A is evaluated using the Spearman Correlation.

This paper makes use of the SAND composition, a set of STR relevant characteristics, as well as regression algorithms trained with these characteristics in order to predict the STR score of other sentence pairs. In this paper the precise characteristics, algorithms, and parameters are presented as well as language based analysis. The final scores correspond to the best performing regression algorithm.

Our work also compares different metrics and their influence on the STR task compared to classic semantic similarity, as well as the model's varying behavior over the different languages used.

The paper is structured as follows: Section 2 explains the theories and models that are the background of our proposal. Section 3 explains the set of characteristics that have been chosen for our experiments. Section 4 explains the configuration of the data and the experimental methodology. In Section 5 we discuss the results obtained in the experiments, and compare them to the scores of other participants in the track. We close in Sec-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1288–1292

tion 6 with some conclusions and ideas for further experiments.

2 Related Work

In the field of STR, various methodologies have been proposed. One such approach, outlined by Asaadi et al. (2019), involves analyzing the relatedness between word bi-grams. They describe the construction of a dataset tailored for this purpose. To compute the STR between bi-grams, or between bi-grams and unigrams, they utilize word embeddings represented as vectors generated by GloVe, fastText, and models employing matrix factorization of word-context co-occurrence matrices. They explore various methods for composing bi-gram vectors, such as addition, multiplication, tensor product with convolution, and dilation. The relatedness between two vectors is determined by computing the cosine similarity between them.

In a study by Abdalla et al. (2023), the concept of Semantic Textual Relatedness (STR) is extended to encompass the comparison between entire sentences. The authors delineate the construction of a specialized dataset tailored for this task and show the annotation process applied to this dataset. Their investigation delves into the influence of various linguistic factors, including lexical overlap, related words, related words belonging to the same part of speech, and the relatedness of subjects or objects, on the semantic relatedness between pairs of sentences. They represent each sentence as a vector and employ cosine similarity between these vectors as a metric for predicting semantic relatedness. To facilitate this analysis, they use static word embeddings such as Word2Vec, GloVe, and fastText, as well as contextual word embeddings like BERT and RoBERTa.

3 SAND Composition

This section describes the SAND (named based on the used characteristics: **S**entence Transformers, **A**nglE Embeddings, **N**-grams, Sentence Length **D**ifference coefficient) regression system used for the task. The STR datasets are comprised of senbtence pairs and a target score, in order to train the model with task relevant information certain similarity metrics were chosen in order to create a vector of characteristics that represent each sentence pair, such vectors were used as training data for the regression algorithms. An initial approach relied on similarity metrics such as Jaccard, and Dice coefficients as well as Jaro-Winkler and Levenshtein distance, nonetheless these metrics proved to be inefficient at training the model adequately, returning poor results when evaluated with partitioned training data. After revising the dataset and the nature of the sentences in question, the initial chosen characteristics were a coefficient analyzing the length of the sentences with and without stoptwords. Consider x, y two sentences, then both coefficients are obtained from the following:

$$\text{LenCoef}(x, y) = |\frac{\text{length}(x) - \text{length}(y)}{\text{length}(x) + \text{length}(y)}| \quad (1)$$

An observation on lexical overlapping in highly related sentences led to the choice of an n-gram based coefficient. For $n \in \{1, 2, 3\}$, the n-gram coefficient is defined as:

$$\text{n-gramCoef}(x, y) = \frac{\text{n-grams}(x) \cap \text{n-grams}(y)}{\text{n-grams}(x)} \quad (2)$$

Sentence pairs in the dataset often rely on contextual similarity apart from lexical overlap when it comes to measuring relatedness, meaning the use of a pretrained model is in order. The first approach relied on the use of Sentence Transformers (ST) (Reimers and Gurevych, 2019), a siamese neural net that uses pretrained encoders in order to generate contextualized sentence embeddings. Each pair of sentences was embedded using the ST architecture and the cosine similarity of the resulting vectors was obtained as the initial pretrained characteristic. Formally speaking:

$$\text{ST}_{sim}(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \, \|y\|} \quad (3)$$

Nonetheless standard encoder generated embeddings have shown to be improvable, for that the final characteristic relies on similarity based on AnglE-optimized embeddings (Li and Li, 2023). These type of embeddings optimize the cosine similarity saturation zones during training using complex space embeddings so that resulting vectors achieve a higher level of similarity. Nonetheless the final comparison between AnglE vectors is done in the same manner as equation 3.

Once these characteristics are extracted from each sentence pair they are passed to various regression algorithms for training, validation and testing. For this task four regression algorithms

were used. The reported scores correspond to the best model for each language. The precise details of the implementation are presented next.

4 Experimental Setup

4.1 Data and Evaluation Methodology

We use the official dataset (Ousidhoum et al., 2024a) in English and Spanish for Task 1, track A (supervised), which is structured as follows: PairID, Text and Score. PairID is an identifier of the pair, the Text column is the sentence pair separated by a line break, and the Score column is a float number between 0 (completely unrelated) and 1 (maximally related) which indicates the degree of semantic textual relatedness between the two sentences.

The English training corpus is composed of 5500 sentence pairs meanwhile the Spanish counterpart has 1561 pairs. The score distribution comparison in figures 1 and 2 indicates that the English scores have a wider variance than the Spanish ones, nonetheless the most represented scores (scores corresponding to over 50 sentence pairs) follow roughly a Gaussian distribution. In contrast the Spanish score distribution doesn't behave as cleanly.

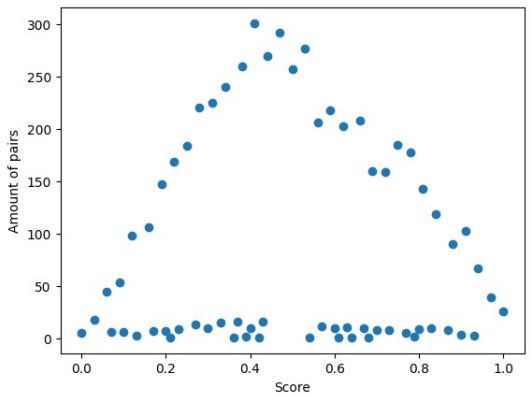


Figure 1: English score distributions.

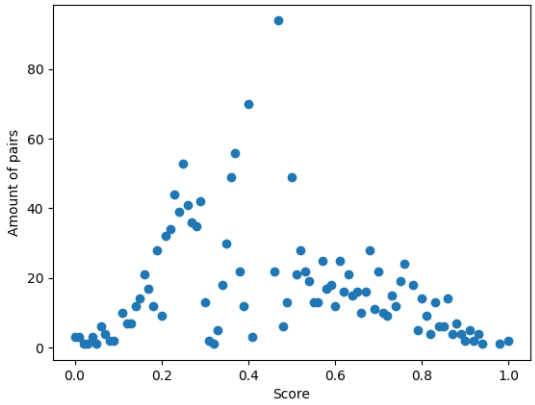


Figure 2: Spanish score distributions.

The results of the shared task are evaluated with the Spearman rank correlation coefficient which is used to discover the strength of a relationship between two sets of data, in this case two sentences.

4.2 Algorithms, parameters and pretrained models

The following regression algorithms, along with the particular parameters, were trained and validated using 7-dimensional vectors corresponding to each characteristic in the SAND Composition:

- SVM and SVM with epsilon: Both algorithms' used the default regularization parameter of 1, meanwhile the SVM ϵ variation used $\epsilon = 0.3$.
- RandomForest: Default parameters were used except for maximum depth which was set to 5, and randomness of the bootstrapping of the samples used when building trees was set at 0.
- Ridge regression: Default parameters except the constant that multiplies the L2 term α. For the English corpus α was set at 0.8, while for the Spanish corpus it was set at 0.9.

Regarding ST and AnglE Embeddings the use of a pretrained model was necessary in order to compute the embeddings and eventually the similarity score. For the English version of ST the ALL-MPNET-BASE-V2 checkpoint was used, meanwhile for the Spanish version the PARAPHRASE-MULTILINGUAL-MPNET-BASE-V2 checkpoint was chosen, both developed particularly for the ST architecture (Reimers and Gurevych, 2020). Meanwhile AnglE embeddings

used the ANGLE-BERT-BASE-UNCASEDNLI-EN-V1 checkpoint, prioritizing the BERT (Devlin et al., 2019) based model over the LLaMA (Touvron et al., 2023) based one due to computational power needed for each model.

5 Results and Analysis

The final SAND composition was the result of an ablation test performed using various combinations of different characteristics. The previously mentioned regression algorithms were trained and tested using each individual characteristic as well as different combinations of each. The SAND composition was the best performing combination for both English and Spanish, each of the four regression algorithms achieved the best result in the ablation test with SAND than with each simpler combination.

Tables 1 and 2 show the best Spearman Correlation for each characteristic combination as well as the model that achieved said score in each language when evaluating in the validation dataset.

Char	SVM	RF	SVM_ϵ	Ridge
ST	0.7891	0.7847	0.7865	0.7891
AnglE	0.7789	0.7737	0.7772	0.7789
N-grams	0.6634	0.6496	0.6620	0.6584
Distance	0.2343	0.2090	0.2839	0.2888
SAND	**0.7986**	**0.8197**	**0.7992**	**0.7921**

Table 1: English Spearman Correlation for different characteristics.

Char	SVM	RF	SVM_ϵ	Ridge
ST	0.6397	0.6058	0.6310	0.6397
AnglE	0.6140	0.5976	0.6038	0.6212
N-grams	0.6425	0.6232	0.6419	0.6431
Distance	0.5595	0.5584	0.5586	0.5594
SAND	**0.688**	**00.6783**	**0.6937**	**0.7029**

Table 2: Spanish Spearman Correlation for different characteristics.

Finally table 3 shows the results of each model with the SAND Composition. The reported results correspond to the predicted scores made by the highlighted models: Random Forest for English, and Ridge Regression for Spanish.

It is important to note that the embeddings created with pretrained models were the feature with the greatest impact on our model. Even as an isolated measure they both prove to be better metrics than their n-grams and distance counterparts. Similarly, considering that both BERT and ALL-MPNET-BASE-V2 are trained in English primarily, it is logical that the regression algorithms performed better in said language.

Model	Spanish	English
RandomForest	0.6968	**0.8197**
SVM	0.6881	0.8133
Ridge	**0.7029**	0.8117
SVM Epsilon	0.6997	0.7945

Table 3: Spearman coefficient for SAND composition.

Nonetheless the SAND Composition proves that these characteristics can be complemented and improved using relevant information such as n-grams coefficients. Since they don't rely on pretrained models and rather focus on lexical overlapping, this coefficient was able to discern certain relatedness measures.

SAND Composition was able to achieve the best results of the ablation test meaning that regression models do benefit from the mix of characteristics and still be relevant in a competition setting.

6 Conclusion

In this paper, we describe the SAND composition for the STR shared task, which is based on both semantic and lexical features, because we observe that: two sentences can share most of the words and apparently have no semantic relation but a high value of Spearman coefficient and vice versa, they can share semantics without matching words. With this in mind the SAND Composition contains half semantic features, and half lexical ones. This approach allowed achieved the 18th place in English and 2nd place in Spanish, with 0.83 and a 0.73 Spearman Correlation score respectively. In both cases the results are over the baseline and only 0.05 of the first place in English and 0.01 in Spanish, meaning SAND proves to have relevant characteristics.

For future experiments added features that consider both semantic, lexical and contextual parts simultaneously might prove to be more efficient than various unrelated metrics. Mixing word embeddings and PoST tagging might generate a relatedness score that proves to be more useful than separate similarity metrics.

Acknowledgments

This research was funded by CONAHCYT (CF-2023-G-64) and PAPIIT project IT100822. G.B.E. is supported by a grant for the requalification of the Spanish university system from the Ministry of Universities of the Government of Spain, financed by the European Union, NextGeneration EU (María Zambrano program, Universitat de Barcelona).

K. Salas-Jimenez thanks CONAHCYT scholarship program (CVU: 1291359).

Ángel Cadena thanks CONAHCYT scholarship program (CVU: 1227093).

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2023. What makes sentences semantically related: A textual relatedness dataset and In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Dubrovnik, Croatia.

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. 2019. Big BiRD: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota. https://doi.org/10.18653/v1/N19-1050.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4171–4186. https://doi.org/10.18653/v1/N19-1423.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871* .

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *CoRR* abs/2004.09813. https://arxiv.org/abs/2004.09813.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Team UTSA-NLP at SemEval 2024 Task 5: Prompt Ensembling for Argument Reasoning in Civil Procedures with GPT4

Dan Schumacher[1] and **Anthony Rios**[2]
[1]MS Data Analytics, College of Business
[2]Department of Information Systems and Cyber Security
The University of Texas at San Antonio
dan.schumacher@my.utsa.edu, anthony.rios@utsa.edu

Abstract

In this paper, we present our system for the SemEval Task 5, *The Legal Argument Reasoning Task in Civil Procedure* Challenge. Legal argument reasoning is an essential skill that all law students must master. Moreover, it is important to develop natural language processing solutions that can reason about a question given terse domain-specific contextual information. Our system explores a prompt-based solution using GPT4 to reason over legal arguments. We also evaluate an ensemble of prompting strategies, including chain-of-thought reasoning and in-context learning. Overall, our system results in a Macro F1 of .8095 on the validation dataset and .7315 (5th out of 21 teams) on the final test set. Code for this project is available at https://github.com/danschumac1/CivilPromptReasoningGPT4.

1 Introduction

Mastering the reasoning behind legal arguments is a fundamental skill required of all law students. In this study, we develop a novel approach for SemEval Task 5, The Legal Argument Reasoning Task in Civil Procedure Challenge (Held and Habernal, 2024). The SemEval Task released a dataset that was scraped from *The Glannon Guide To Civil Procedure*, a textbook designed for law students. Specifically, given case law, a question, and a potential answer to that question, students must be able to reason over the contextual information (case law) to determine if the question is correct or not.

There has been substantial research in developing NLP-based reasoning systems (Guha et al., 2024; Bongard et al., 2022; Chalkidis, 2023; Blair-Stanek et al., 2023; Kuppa et al., 2023; Yu et al., 2022). The methods can be categorized into two major frameworks: fine-tuning and large language model-based (LLM) solutions. For fine-tuning approaches, Bongard et al. (2022) introduced an approach that fine-tunes LegalBERT (Chalkidis et al., 2020) and developed several methods for handling long text that does not fit within the token limitations of LegalBERT. For LLM solutions, Chalkidis (2023) explored the use of ChatGPT for solving legal exams. Guha et al. (2024) introduced a more comprehensive evaluation benchmark geared to large language models consisting of 162 tasks. Many of their experiments show that GPT4 is one of the top performing approaches for legal reasoning across all language models, while Flan-T5-XXL is the best open-source option. Although showing substantial generalization is important, developing specific prompting strategies for different reasoning tasks can substantially improve performance. Hence, this paper adds to existing literature on the exploration of prompting approaches in the legal domain.

For this work, we adapt several prompting-based strategies to develop an LLM-based solution for the shared task. Specifically, we combine a retrieval system with in-context learning and chain-of-thought reasoning. There are several studies showcasing the utility of in-context learning (Liu et al., 2022a; Lu et al., 2022) and chain-of-thought reasoning (Wei et al., 2022; Sun et al., 2023; Yao et al., 2024). Moreover, there is work using both human-curated and machine-generated reasons. In this work, we focus on human-generated reasons for the training data, and machine-generated examples are only used at test times when human expert annotations are not provided. Furthermore, rather than providing a step-by-step reasoning approach, which may not make sense in this context, our approach is more similar to single-step rationales (Brinner and Zarrieß, 2023; Yasunaga et al., 2023), which simply provides a single step of reasoning for why an answer is correct or incorrect.

In summary, this paper makes the following contributions to our solution for the SemEval 2024 Task 5 shared task:

- We evaluate prompting strategies using GPT4

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1293–1301

that combine several popular ideas, including in-context learning and chain-of-thought reasoning.

- In-context learning can be sensitive to the actual choice of examples, particularly when only a few examples are provided. Hence, we also explore an ensemble of prompt-based predictions to improve overall performance.

- Finally, we provide a unique error analysis where we found limitations and common error types generated by GPT4 using our prompting strategies. For example, when a part of an answer candidate is correct, but the reasoning is wrong, GPT4 is likely to generate a false positive (i.e., predict it is correct instead of incorrect).

2 RELATED WORK

Overall, there are three major areas of legal NLP: legal question-answering (Khazaeli et al., 2021; Kien et al., 2020; Ryu et al., 2023; Martinez-Gil, 2023; Wang et al., 2023), judgment prediction (Masala et al., 2021; Valvoda et al., 2023; Juan et al., 2023), and corpus mining (e.g., summarization, text classification, information extraction, and retrieval-related research) (Poudyal et al., 2020; Li et al., 2022; Vihikan et al., 2021; Zhang et al., 2023; Limsopatham, 2021; de Andrade and Becker, 2023). There has also been some broad methodology work that is aimed at working on various legal tasks in general (e.g., LegalBERT (Chalkidis et al., 2020)).

The SemEval task is most similar to question-answering related research. In the domain of legal question-answering, recent research efforts are focused on creating new systems, developing evaluation criteria, and compiling datasets, considering the significant variation across different legal fields. Khazaeli et al. (2021) introduced a commercial question-answering system for legal inquiries, leveraging information retrieval techniques, sparse vector search, embeddings, and a BERT-based reranking system, trained on both general and legal domain data. Ryu et al. (2023) developed a novel evaluation method for LLM-generated texts that assess their validity using retrieval-augmented generation, showing improved alignment with legal experts' assessments and effectiveness in identifying factual errors. Wang et al. (2023) created the Merger Agreement Understanding Dataset (MAUD), a unique, expert-annotated dataset for legal text reading comprehension, highlighting promising model performance and the need for further improvement in understanding complex legal documents.

From a methodological point-of-view instead of a task-oriented view, recent research efforts have concentrated on the advancement of NLP-based reasoning systems, with a particular focus on applications within the legal domain (Guha et al., 2024; Bongard et al., 2022; Chalkidis, 2023; Blair-Stanek et al., 2023; Kuppa et al., 2023; Yu et al., 2022). These efforts can be broadly classified into two distinct methodologies: fine-tuning approaches and those leveraging large language models (LLMs). Within the fine-tuning paradigm, Bongard et al. (2022) have proposed modifications to LegalBERT (Chalkidis et al., 2020) aimed at enhancing its ability to process texts that exceed the model's inherent token limitations. This approach is representative of a broader trend towards tailoring pre-existing models to better suit specific textual analysis tasks in the legal sector. Kien et al. (2020) developed a retrieval-based model employing neural attentive text representation with convolutional neural networks and attention mechanisms for accurately matching legal questions to relevant articles, demonstrating superior performance on a Vietnamese legal question dataset.

Conversely, the exploration of LLMs has also been wide covering new general approaches for legal question answering and reasoning to new datasets and benchmarks. Yu et al. (2023) investigated the impact of chain-of-thought prompts and fine-tuning methods on legal reasoning tasks, specifically the COLIEE entailment task, and found that prompts based on legal reasoning techniques and few-shot learning with clustered training data significantly enhance performance. Chalkidis (2023) demonstrates the potential of utilizing models like ChatGPT for complex reasoning tasks, such as solving legal examination questions. Building on this, Guha et al. (2024) have introduced a comprehensive evaluation framework designed specifically for assessing the capabilities of LLMs across a suite of 162 tasks. This benchmark aims to provide a more nuanced understanding of the strengths and limitations of LLMs in the context of legal reasoning. Additionally, while employed within a distinct domain from legal reasoning, a comparable methodology in prompt engineering—encompassing chain of thought prompting

and in-context prompting—is illustrated in Liu et al.'s recent work (Liu et al., 2022b). Overall, compared to the prior work that developed methods and datasets for generating answers to questions, the SemEval task focuses on understanding whether a provided answer candidate is valid given a specific context.

3 METHOD

We provide a high-level overview of our approach in Figure 1. Overall, we explore three major different prompting approaches: Zero-shot prompting, few-shot prompting, and few-shot prompting with chain-of-thought-like reasoning. Moreover, we explore an ensemble of multiple approaches. The methods are described in the following subsections.

3.1 Few-Shot Retrieval Augmented Chain-of-Thought Prompting

We refer to our approach as "Few-Shot & CoT & RAG." Each example within the dataset contains an Introduction, Question, Answer Candidate, Analysis, and Label. For new examples at test time, we only have access to the Introduction, Question, and Answer Candidate. The Introduction consists of a general background about a legal case. The Question is about the case, and the Answer Candidate is an answer to the Question. It is important to note that the Question could be in question form, where the answer directly answers what is asked. However, it may function as a fill-in-the-blank exercise, where the question presents an incomplete statement that the Answer Candidate is expected to complete. The Analysis, which is only provided in the training and development datasets, is a detailed expert-defined explanation for why the Answer Candidate is or is not valid. The Label is a TRUE or FALSE value, where TRUE means that the Answer Candidate correctly addresses the question given the provided context. Likewise, FALSE means that the Answer Candidate is incorrect.

As shown in Figure 1, our prompting strategy contains three main components, a system prompt, the in-context examples, and the final test instance we will classify as TRUE or FALSE. The system prompt describes what the large language model (LLM) should do. In the Figure, it is shown that we also added explicit information to limit the model from generating non-relevant information.

The in-context examples are provided to the LLM before the final text instance. Intuitively, the goal is to provide some examples of the task we are accomplishing to better ground the LLM to make better-generated responses. This is the Retrieval Augmented aspect of our system (i.e., RAG). While any random examples could be provided, we search for the most relevant examples for each test instance. Formally, given an input instance x_i consisting of a concatenated Introduction, Question, and Answer triplet w, we retrieve the most similar examples $\{x_1, \ldots, x_{\mathcal{N}(x_i)}\}$, where $\mathcal{N}(x_i)$ are the k most similar examples to x_i. Each question-answer pair is embedded using LegalBERT. The in-context examples all come from the provided training dataset. Once retrieved, all of the relevant information for the retrieved examples are used in the prompt (i.e., the Introduction, Question, Answer Candidate, Analysis, and Label). Figure 1 shows an example with two in-context examples.

Finally, the last component of the prompt is the text example. Basically, for every example we wish to make a prediction for, we pass the Introduction, Question, and Answer Candidate. The model will first generate the Analysis for that example, then it will generate the Label.

3.2 Ensemble

Besides the method described in the previous section, we also explored prompting variants to create an ensemble. We describe each of the additional methods below (besides the Few-SHOT & CoT & RAG method described in the previous subsection).

Zero-shot. This approach does not use any in-context examples. We provide the system prompt and the test example to the model directly to get the final prediction.

Zero-shot & CoT. This method builds on the Zero-Shot approach by adding the CoT aspect to the test example. Note that this is not available at test time, so the model generates an Analysis section without previous examples.

Few-Shot. The few-shot approach will use multiple in-context examples. However, unlike our main method explained in the previous section, it uses the same in-context examples for all test cases. The examples were chosen in an ad-hoc manner with an emphasis on relatively short examples to limit the number of tokens to reduce costs.

Few-Shot & CoT. This builds on the Few-Shot method, where ad-hoc in-context examples are still

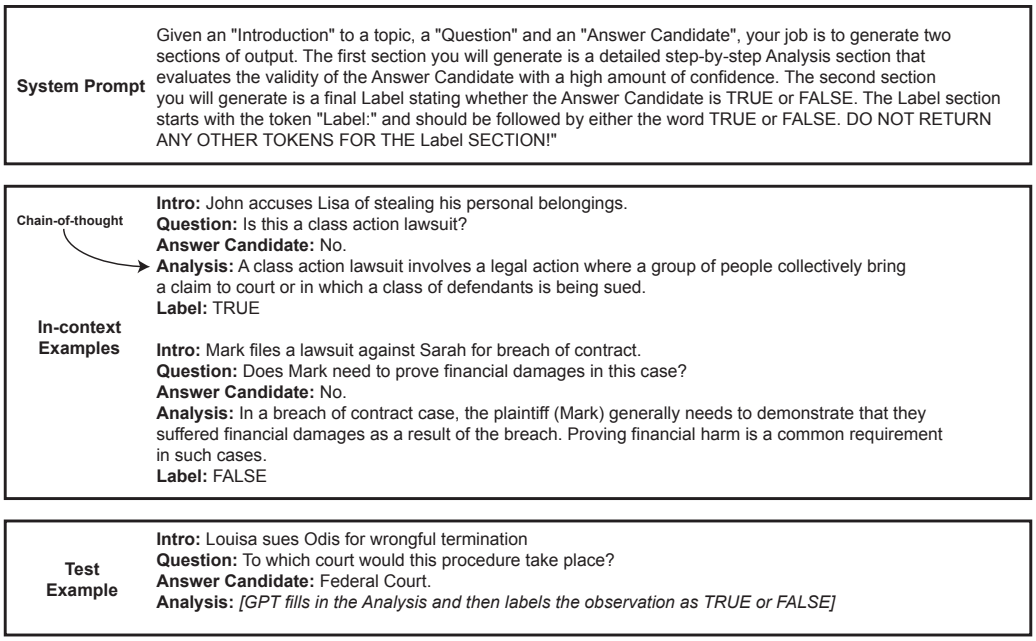


Figure 1: Overview of our prompting strategy.

used, but we also provide the CoT reasoning (Analysis) section to the in-context examples. The GPT4 model will generate the Analysis section for each test example.

Few-Shot & RAG. This also builds on the Few-Shot method, but instead of using ad-hoc examples, it uses LegalBERT and cosine similarity to find relevant in-context examples for each test case (as explained in the previous section).

Overall, there are a total of 4 models in our ensemble. Few-Shot and Few-Shot & RAG were not used, but we evaluated them. The models in the ensemble were chosen by checking combinations on the validation dataset. To make a prediction, we use voting with a threshold (i.e., where the votes are processed to generate the proportion of TRUE values).

3.3 Model Details

For all of our prompts, we use GPT-4-1106-preview with a temperature of .7. The similarity metric used for finding relevant in-context examples is cosine similarity. Moreover, we choose a total of 2 in-context examples, consisting of 1 TRUE example and 1 FALSE example. We searched for the best threshold after voting by calculating the proportion of TRUE vs. FALSE predictions, ultimately choosing .5 as the threshold.

There were many cases where GPT4 does not return a final label in an easy-to-process fashion (i.e., it does not end with a TRUE or FALSE). We explored multiple approaches to parse the answer. Our ultimate strategy involves re-submitting examples to GPT-4 that initially failed to produce a valid label, explicitly indicating the absence of the label, and prompting the model to generate the correct information.

Method	Macro F1	Acc.
Baselines		
RoBERTa	.5128	.2286
Legal-BERT	.5575	.2941
Zero-shot	.6681	.7857
Zero-Shot & COT	.7162	.7500
Few-shot	.6935	.7738
Few-shot & COT	.6762	.7262
Few-shot & RAG	.6898	.7500
Few-Shot & COT & RAG (Ours)	.7306	.7857
Ensemble (Ours)	.8095	.8571

Table 1: Validation dataset results

4 RESULTS

In this section, we present our own results on the validation dataset as well as the final results in the competition.

Baselines In our experiments, using the validation data, we compare our approach (Few-SHOT & CoT & RAG) with the other approaches used in the Ensemble. Moreover, we compare our system to both RoBERTa (Liu et al., 2019) and Legal-BERT. However, because both RoBERTa and Legal-BERT are limited to 512 tokens, we split the Introduction into b pieces. b is calculated by subtracting the number

of words in the question and answer from 512. We halve that result to accommodate the fact that the number of tokens typically exceeds the number of words. Next, we divide the number of words in the Introduction by the previously calculated number to obtain the total number of windows. Finally, all of the tokens in the Introduction are evenly split into the windows. Each piece of the introduction is appended to the Question and Answer pair independently to generate multiple predictions for each instance. We then use voting to make a final prediction for the entire sequence. This method is similar to what was explored in Bongard et al. (2022).

Validation Results. The validation performances are shown in Table 1. We observe that the fine-tuned methods (e.g., RoBERTa), perform less effectively compared to all methods utilizing GPT-4. Between the fine-tuned methods, we find that Legal-BERT outperforms RoBERTa, which is expected given Legal-BERT was fine-tuned on relevant corpora.

Next, between GPT4 methods (not including the ensemble), we find that performance varies substantially between .6681 and .7162 for Macro F1. All methods outperform Zero-Shot. However, Zero-Shot & COT achieved the best performance across all baseline methods for Macro-F1. When we compare the baseline approaches to our method (Few-Shot & COT & RAG), we find that the method outperforms all variations. From an ablation standpoint, removing RAG has the biggest performance drop (.7306 vs. .6762). Interestingly, we find that removing CoT has the second largest drop in performance (.7306 vs. .6935), and removing the in-context examples has the smallest drop in performance (.7306 vs. .7162). Before the study, we expected removing the in-context examples would result in the largest performance drop.

Competition Results. In Table 2, we report the final results of the competition, achieving a Macro F1 of .7315. But, why do we see such a large performance drop between the competition and validation results (.7315 vs. .8095)? We hypothesize two major reasons. First, we realized that the validation dataset contains many Introduction-Question pairs identical to the ones in the training dataset. Despite the Answer Candidates differing across the two datasets, the substantial overlap in Introduction-Question pairs may lead to an overestimation of our model's performance on the validation dataset,

#	User	Macro F1	Acc.
1	zhaoxf4	.8231	.8673
2	irene.benedetto	.7747	.8265
3	kbkrumov	.7728	.8367
4	qiaoxiaosong	.7644	.8163
5	**UTSA-NLP**	**.7315**	**.7959**
6	kubapok	.6971	.7857
7	samyak	.6599	.7449
8	hrandria	.6327	.6939
9	Yuan_Lu	.6000	.6327
10	PengShi	.5910	.6735
11	msiino	.5597	.5714
12	Hwan_Chang	.5556	.5918
13	kriti7	.5511	.6020
14	woody	.5510	.6633
15	odysseas_aueb	.5143	.6122
16	Manvith_Prabhu	.4966	.6224
17	lhoorie	.4957	.5000
18	yms	.4827	.7245
19	U_201060	.4503	.6633
20	langml	.4375	.4490
21	lena.held	.4269	.7449

Table 2: Final Competition Results. Our submission is in **bold** font.

	predFalse	**predTrue**
actFalse	57	9
actTrue	3	15

Table 3: Confusion matrix for the validation dataset. actFalse and actTrue stand for actual True and False values, respectively. predFalse and predTrue stand for predicted False and predicted True.

rendering it potentially too optimistic when applied to entirely new data. Second, we spent time over-optimizing the ensemble on the validation dataset causing overfitting issues (e.g., checking thresholds, model combinations, and more). By generating a better validation split, we may have seen better generalization.

Error Analysis Our method resulted in twelve mistakes on the validation dataset: ten false positives and two false negatives. The confusion matrix is shown in Figure 3. In Table 4 (See Appendix), we categorize each false positive into one of four categories: "Incorrect reasoning," "Shared the same introduction and question pair," "lots of similar language", and "Other." With only two false negatives, there are few useful patterns to understand among the errors. Hence, we only have a general FN category. However, from the false positives, we make two major findings which we describe below.

The first was when the answer candidate had the correct answer but *incorrect reasoning*. Here is a toy example demonstrating this pattern:

> **Introduction**: Carlos enjoys riding his skateboard in the skate park. Unfortunately, during one of his rides, he fell and split his head open.
>
> **Question**: Should Carlos go to the hospital?
>
> **Answer Candidate**: Yes, Carlos should go to the hospital because he likes to kick-flip so much.
>
> **Analysis**: While it is correct that Carlos should go to the hospital for medical attention after splitting his head open, the reasoning provided in the answer candidate is flawed. The decision to seek medical help should be based on the severity of the injury and the need for professional medical treatment, not on Carlos's enjoyment of skateboard stunts.

Intuitively, part of the Answer Candidate is correct, i.e., Carlos *should* go to the hospital, yet the reasoning that states why he should go to the hospital is wrong.

The second point is that three of our ten false positives all *shared the same introduction and question pair.* The introduction contained more extraneous information than usual and was 128 words longer than the average. In these instances, our model would analyze the answer candidate as correct but without taking into account the particular case that was asked about. Below is a toy example:

> **Introduction**: My dog Louisa loves to learn new tricks, go for walks, eat her dinner, then sleep through the night
>
> **Question**: What does Louisa like to do *after dinner*?
>
> **Answer Candidates**:
>
> 1. Learn new tricks (*wrong*)
> 2. Go for walks (*wrong*)
> 3. Eat her dinner (*wrong*)
> 4. Sleep through the night (*correct*)

In the example, three of the answers are incorrect, while "Sleep through the night" is the correct example. When the Introduction is long, the actual context of the question may be ignored, which can be interpreted as a needle in a haystack issue. Developing better systems that provide direct "attention" to relevant information may improve performance.

Finally, a third point that we believe caused our model to predict incorrect answer candidates is *similar language* in both the introduction and question compared to the answer candidate.

> **Introduction:** While making eggplant Parmesan for the first time in a buttery dutch over Paige burned herself on the stove
>
> **Question:** How does Paige remember this incident.
>
> **Answer Candidates:** She remembers making eggplant Parmesan for the first time in a buttery dutch oven fondly.

5 CONCLUSION

In this paper, we described our approach for 2024 SemEval Task 4, The Legal Argument Reasoning Task in Civil Procedures. Specifically, we introduced a GPT4 prompting-based strategy that achieved 5th place in the competition out of 21 participants. Overall, we find that combining in-context learning, where we use a retrieval-based approach to find relevant examples, as well as in-context learning improves model performance. Based on our experiments, there are three natural areas for future research. First, the actual Analysis section used for chain-of-thought reasoning does not match traditional methods which use step-by-step reasoning. Hence, a logical next extension is to reword (potentially with GPT4) the Analysis section to provide a step-by-step explanation for an answer. Second, this work was limited to 2 in-context examples to limit API costs and allow us to test other models in our initial experiments. However, extending that to 10 or more examples can potentially improve performance. Third, the current approach relies on a closed-source model (GPT4). Exploring open-source models, particularly smaller open-source models such as T5 (Chung et al., 2022) and LLama2 (Touvron et al., 2023), is important to better understand the impact of pretraining data on performance.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 2145357.

References

Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can gpt-3 perform statutory reasoning? *arXiv preprint arXiv:2302.06100*.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207.

Marc Brinner and Sina Zarrieß. 2023. Model interpretability and rationale extraction by input mask optimization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13722–13744.

Ilias Chalkidis. 2023. Chatgpt may pass the bar exam soon, but has a long way to go for the lexglue benchmark. *arXiv preprint arXiv:2304.12202*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Leonardo de Andrade and Karin Becker. 2023. Bb25hlegalsum: Leveraging bm25 and bert-based clustering for the summarization of legal documents. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 255–263.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. 2024. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36.

Lena Held and Ivan Habernal. 2024. SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Yining Juan, Chung-Chi Chen, Hsin-Hsi Chen, and Daw-Wei Wang. 2023. Custodiai: A system for predicting child custody outcomes. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 10–16.

Soha Khazaeli, Janardhana Punuru, Chad Morris, Sanjay Sharma, Bert Staub, Michael Cole, Sunny Chiu-Webster, and Dhruv Sakalley. 2021. A free format legal question answering system. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 107–113.

Phi Manh Kien, Ha-Thanh Nguyen, Ngo Xuan Bach, Vu Tran, Minh Le Nguyen, and Tu Minh Phuong. 2020. Answering legal questions by learning neural attentive text representation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 988–998.

Aditya Kuppa, Nikon Rasumov-Rahe, and Marc Voses. 2023. Chain of reference prompting helps llm to think like a lawyer. In *Generative AI+ Law Workshop*.

Xiang Li, Jiaxun Gao, Diana Inkpen, and Wolfgang Alschner. 2022. Detecting relevant differences between similar legal texts. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 256–264.

Nut Limsopatham. 2021. Effectively leveraging bert for legal document classification. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 210–216.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Yilun Liu, Shimin Tao, Weibin Meng, Jingyu Wang, Wenbing Ma, Yuhang Chen, Yanqing Zhao, Hao Yang, and Yanfei Jiang. 2022b. Interpretable online log analysis using large language models with prompt strategies. In *Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (Wordplay 2022)*, Seattle, United States. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*.

Jorge Martinez-Gil. 2023. A survey on legal question–answering systems. *Computer Science Review*, 48:100552.

Mihai Masala, Radu Cristian Alexandru Iacob, Ana Sabina Uban, Marina Cidota, Horia Velicu, Traian Rebedea, and Marius Popescu. 2021. jurbert: A romanian bert model for legal judgement prediction. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 86–94.

Prakash Poudyal, Jaromír Šavelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. Echr: Legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75.

Cheol Ryu, Seolhwa Lee, Subeen Pang, Chanyeol Choi, Hojun Choi, Myeonggee Min, and Jy-Yong Sohn. 2023. Retrieval-based evaluation for llms: A case study in korean legal qa. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 132–137.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. *arXiv preprint arXiv:2305.08377*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Josef Valvoda, Ryan Cotterell, and Simone Teufel. 2023. On the role of negative precedent in legal outcome prediction. *Transactions of the Association for Computational Linguistics*, 11:34–48.

Wayan Oger Vihikan, Meladel Mistica, Inbar Levy, Andrew Christie, and Timothy Baldwin. 2021. Automatic resolution of domain name disputes. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 228–238.

Steven H Wang, Antoine Scardigli, Leonard Tang, Wei Chen, Dimitry Levkin, Anya Chen, Spencer Ball, Thomas Woodside, Oliver Zhang, and Dan Hendrycks. 2023. Maud: An expert-annotated legal nlp dataset for merger agreement understanding. *arXiv preprint arXiv:2301.00876*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H Chi, and Denny Zhou. 2023. Large language models as analogical reasoners. *arXiv preprint arXiv:2310.01714*.

Fangyi Yu, Lee Quartey, and Frank Schilder. 2022. Legal prompting: Teaching a language model to think like a lawyer. *arXiv preprint arXiv:2212.01326*.

Fangyi Yu, Lee Quartey, and Frank Schilder. 2023. Exploring the effectiveness of prompt engineering for legal reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13582–13596.

Ruizhe Zhang, Qingyao Ai, Yueyue Wu, Yixiao Ma, and Yiqun Liu. 2023. Diverse legal case search. *arXiv preprint arXiv:2301.12504*.

A Error Analysis

Table 4 reports the basic statistics for the error types we observed in our error analysis.

Type	Frequency	Example
FP: Incorrect Reasoning	16.67% (1/12)	Carlos enjoys riding his skateboard in the skate park...
FP: Shared the same introduction and question pair.	25% (3/12)	My dog Louisa loves to learn new tricks...
FP: Lots of Similar Language	8.33% (2/12)	While making Eggplant Parmesean...
FP: Other	25% (3/12)	...
FN	25% (3/12)	...

Table 4: Manual categorization of error types. False positives are categorized as either "Incorrect Reasoning," "Shared the same introduction and question pair," "lots of similar language," or "Other." We use a single FN category because of the lack of errors to analyze.

BD-NLP at SemEval-2024 Task 2: Investigating Generative and Discriminative Models for Clinical Inference with Knowledge Augmentation

Shantanu Nath
University of Trento, Italy
shantanu.nath@studenti.unitn.it

Ahnaf Mozib Samin
Queen's University, Canada
ahnaf.samin@queensu.ca

Abstract

Healthcare professionals rely on evidence from Clinical Trial Records (CTRs) to devise treatment plans. However, the increasing quantity of CTRs poses challenges in efficiently assimilating the latest evidence to provide personalized evidence-based care. In this paper, we present our solution to the SemEval- 2024 Task 2 titled "Safe Biomedical Natural Language Inference for Clinical Trials". Given a statement and one/two CTRs as inputs, the task is to determine whether or not the statement entails or contradicts the CTRs. We explore both generative and discriminative large language models (LLM) to investigate their performance for clinical inference. Moreover, we contrast the general-purpose LLMs with the ones specifically tailored for the clinical domain to study the potential advantage in mitigating distributional shifts. Furthermore, the benefit of augmenting additional knowledge within the prompt is examined in this work. Our empirical study suggests that DeBERTa-lg, a discriminative task-specific natural language inference model, obtains the highest F1 score of 0.77 and consistency score of 0.76 on the test set, securing the fourth rank on the leaderboard. Intriguingly, the augmentation of knowledge yields subpar results across most cases.

1 Introduction

Clinical trials are conducted on human subjects to test the safety and effectiveness of the medicine prior to designing a new treatment, especially in evidence-based treatments (Avis et al., 2006). Medical professionals prescribe and treat their patients based on clinical trial reports (CTR) in which the methodology and results of clinical trials are outlined. However, the increasing quantity of CTRs poses a challenge for healthcare professionals to manually assess all of them since this process is both time-consuming and labor-intensive (Bastian et al., 2010; DeYoung et al., 2020).

To tackle the aforementioned issue, recent advancements in natural language processing (NLP) encourage medical professionals to employ large language models (LLMs) to interpret and retrieve medical evidence from large quantities of CTRs (Lee et al., 2020). Employing natural language inference (NLI) and textual entailment in the clinical domain (Bowman et al., 2015), professionals can formulate a prompt or statement, for example, *"A minimum bodyweight of 55kg is required to participate in the primary trial."* and input it along with the CTRs into an LLM to determine whether the statement entails or contradicts the evidence from CTRs (Jullien et al., 2023b). Additionally, LLMs can aid in retrieving evidence from the vast amount of CTRs. This technology has the potential to ensure a higher level of precision and efficiency in delivering personalized evidence-based care.

While LLMs have demonstrated significant performance in numerous NLP tasks in recent years (Brown et al., 2020), it is still challenging for them to be deployed in the clinical domain due to their limitation in semantic and quantitative reasoning in language understanding. Moreover, the distribution shift in the clinical domain makes it even more intricate, which requires extensive research in this field (Miller et al., 2020). To address the challenges, Jullien et al. (2024) organizes the SemEval-2024 Task 2, titled "Safe Biomedical Natural Language Inference for Clinical Trials", which aims to investigate the robustness of NLI models when applied to clinical trials with cancer patients. This task seeks to develop an NLI system to connect the new evidence and infer the knowledge from CTRs to find an inferential relation, namely either entailment or contradiction, between a clinical trial document and a statement/claim.

In this study, we investigate the performance of discriminative and generative transformer-based LLMs in the realm of clinical inference. In addition, we explore the potential of clinical domain-specific

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1302–1308

LLMs and compare them with the general-purpose ones with the hypothesis that LLMs pre-trained on clinical data may exhibit superior performance. Furthermore, we study the impact of augmenting knowledge on the semantic reasoning abilities of these LLMs. Our extensive experiments demonstrate that our system, leveraging the discriminative, general-purpose DeBERTa-lg NLI model, achieves an F1 score of 0.77 and consistency score of 0.76 without employing knowledge augmentation on the test set and ranks fourth on the official leaderboard.

2 Background

2.1 Task Definition

In the textual entailment identification task, each input data contains a medical statement, a section name indicating which section the statement claims about, and one or two CTR records that serve as evidence to verify the statement. If the statement only makes claims about one certain trial defined as a primary trial, then only the primary trial will be used as input data. On the other hand, if the statement claims a comparison between a primary trial and a second trial defined as a secondary trial, both CTRs need to be considered as input text. Table 1 presents an example of CTR with four sections. The task is to determine the inferential relation between the medical statement and the associated section in the CTR(s). There are two possible inferential relations for each statement: entailment and contradiction. Models are designed to predict whether each statement entails or contradicts the associated section from the claimed CTR(s).

2.2 Dataset

The Natural Language Inference (NLI) task is designed based on breast cancer CTRs collected by clinical domain experts (Jullien et al., 2023a). Each CTR dataset consists of four sections: intervention, eligibility criteria, results, and adverse events. Each section contains multiple sentences, Formally, $S_t = s_t^1, s_t^2, ..., s_t^n$, here t denotes for the type of section. The participants are provided a text file containing a statement, 1-2 CTRs, an inference label (Entailment or Contradiction), section that is used for the statement. A statement can be made from a single CTR or a comparison between two CTRs.

3 System overview

In this section we describe about our system for this task.

3.1 Input Prompt

According to the data description, a section contains multiple sentences namely evidence. For short input, we consider only the selected sentences of the section which are annotated as related to the statement. All the sentences are concatenated by adding a space in between each sentence to consider a hypothesis. To design the input text, we consider the statement as premise followed by all the selected sentences as claims from the section of the claimed CTR. A separation token, denoted as [SEP], is used between the statement and the claims. For comparison between two claims, we concatenate the selected sentences from both primary and secondary trials, formally, $C^1 s_t^1, .., C^1 s_t^n, ..., C^2 s_t^1, .., C^2 s_t^n$.

3.2 Knowledge Augmentation

In knowledge augmentation, we consider all the sections as evidence. To design input text, we first take all sentences from the related section as a priority. Then, we concatenated the text with other sentences from the rest of the sections. During the comparison between the two trials, we consider the first 500 tokens from the primary trial and 500 tokens from the secondary trial to limit the length of the sentence to 1024 tokens.

In both cases, we design the prompt in the following way:

$$statement\ [SEP]\ primary\ trial:\ C^1 s_t n.$$
$$secondary\ trial:\ C^2 s_n$$

Secondary trials are added only for comparison between two trials.

3.3 Discriminative Models

Discriminative models, in contrast, are focused on learning the decision boundary that separates different classes within the input data. Instead of modeling the entire data distribution, they concentrate on capturing the conditional probability distribution of labels given the input data. We experiment with a collection of transformer-based discriminative pre-trained language models. We choose models that are trained on medical data, such as electronic

Section	Subsection	Sentence
Intervention	N/A	TX/Maintenance Therapy for Stage IIIB/IV Breast Cancer busulfan: Given orally tamoxifen citrate: Given orally
Eligibility	Inclusion	Hepatic function: Bilirubin =< 2 mg% Karnofsky performance status > 60 Creatinine =< 2.0 mg/dl
	Exclusion	Patient is pregnant Are > 100 days from transplant Are on steroids
Results	Outcome Measurement	Event-free Survival Time frame: 11 years
	Results 1	busulfan: Given orally Overall Number of Participants Analyzed: 50
Adverse events	N/A	Total: 2/50 (4.00%) Pulmonary Emboli [2]1/50 (2.00%)

Table 1: An example of a clinical trial record (shortened) containing four sections namely intervention, eligibility, results and adverse events.

health records, biomedical texts, and scientific articles. We used the BioLinkBERT (Yasunaga et al., 2022) model, which was trained on PubMed abstracts along with citation link information. ClinicalBERT (Wang et al., 2023) trained on a large multicenter dataset with a large corpus of 1.2B words of diverse diseases and utilized a large-scale corpus of EHRs from over 3 million patient records to fine-tune the base language model. Bio_ClinicalBERT (Wang et al., 2023), a domain-specific BERT-based model initialized with Bio-BERT model and fine-tuned with electronic health records from ICU patients, namely MIMIC (Johnson et al., 2016). We also choose a task-specific model, proposed by Laurer et al. (2024), based on DeBERTa large and trained on general domain datasets such as MultiNLI (Williams et al., 2018), Fever-NLI (Nie et al., 2019), Adversarial-NLI (ANLI) (Nie et al., 2020), LingNLI (Parrish et al., 2021) and WANLI (Liu et al., 2022) datasets, which comprise 885,242 NLI hypothesis-premise pairs. A classification layer is added on top of the pre-trained layers and fine-tuned on the training set to predict the probability of entailment or contradiction of the statement. An overall descriptions of the models are provided in table 2.

3.4 Generative Models

For comparison, we also solve this task by using generative models. These models use encoder-decoder architecture to encode input text and directly generate output label entailment/contradiction. Similar to discriminative models, we choose SciFive (Phan et al., 2021) as a domain-specific generative pre-trained model that follows The text-to-text transfer transformer (T5) model (Raffel et al., 2019) and sequence-to-sequence encoder-decoder framework. Pubmed and PMC datasets are utilized for training the models and MIMIC is employed to fine-tune for NLI task. To demonstrate the effectiveness of further fine-tuning the Clinical Trial dataset, we apply both zero-shot and few-shot learning approaches on SciFive. On the other hand, we choose Flan-T5 (Chung et al., 2022) as a general domain generative model. Similar to SciFive, Flan-T5 builds based on T5 architecture. For generative models, we slightly change the prompt. For SciFive, $mednli : sentence1 : < premise > sentence2 : < claims >$ and for FlanT5, $natural\ language; inference : premise : < premise > hypothesis : < claims >$.

Type	Model	Parameters	Variation	Task-specific	Domain-specific
Discriminative	ClinicalBERT	110M	base	No	Yes
	BioLinkBERT	340M	large	No	Yes
	BioClinicalBERT	110M	base	No	Yes
	DeBERTa-lg	304M	large	Yes	No
Generative	FlanT5	250M	base	No	No
	SciFive	770M	large	Yes	Yes

Table 2: Model specifics including the number of trainable parameters in million, variation/size, task-specificity (whether further pre-trained on NLI task or not), and domain-specificity (whether pre-trained on medical domain datasets or not) are shown.

Type	Model	W/o knowledge augmentation			With knowledge augmentation		
		Basline F1	Faithfullness	Consistency	Basline F1	Faithfullness	Consistency
Discriminative	ClinicalBERT	0.56	0.35	0.46	0.54	0.35	0.41
	BioLinkBERT	0.57	0.31	0.49	0.57	0.31	0.49
	BioClinicalBERT	0.58	0.47	0.61	0.57	0.31	0.49
	DeBERTa-lg	0.77	0.80	**0.76**	0.75	0.79	0.75
Generative	FlanT5-base	0.58	0.57	0.63	0.51	0.63	0.61
	SciFive (without FFT)	0.49	0.61	0.49	0.47	0.64	0.51
	SciFive (with FFT)	0.44	0.76	0.63	0.65	0.49	0.62

Table 3: Experiment results of the clinical NLI task on the test set for several discriminative and generative models. We report baseline F1, faithfulness, and consistency scores proposed by (Jullien et al., 2024) for each model with/without knowledge augmentation. The best performance with respect to consistency score is bold-faced. Discriminative DeBERTa-lg achieves the best performance while generative models show promise in several cases. Knowledge augmentation implies including all evidence from CTR concatenated with the prompt statement and then passed as input to the LLM. However, Knowledge augmentation shows negligible impact on model performance. FFT stands for further fine-tuning.

4 Experimental setup

We keep the original data split (1700: 200: 5500) provided by the task organizer for training, validation, and testing sets respectively. Huggingface Transformers [1] library is used for tokenization and further finetuning. Data preprocessing steps are mainly adapted from Vladika and Matthes (2023). For short text, the max sequence length for the tokenizer is set to 256 and 512 for long text for BioLinkBERT, ClinicalBERT and BioClinicalBERT. For Deberta-large, SciFive and Flan-T5 the max sequence length for the tokenizer is set to 512 and 1024 for short and long input text respectively. We train all language models for 20 epochs and an AdamW (Loshchilov and Hutter, 2019) optimizer is used for optimization with a default learning rate of 5e-6 for discriminative models and 5e-5 for generative models with weight ratio of 0.06 and weight decay of 0.01. The models are evaluated on the validation set after each epoch by using precision, recall, and F1 scores and saved best model based on least evaluation loss. We measure the performance of the models based on Faithfulness and Consistency proposed by (Jullien et al., 2024). Faithfulness measures the ability to predict the output based on the correct reason. Therefore, if semantic reason change in future, models will be able to change its prediction accordingly. On one hand, consistency measures the ability to make same prediction for the semantically equal statements which ensures the semantic preserving in a model.

5 Results and Discussion

The experimental results are presented in Table 3. All results are calculated on the standard test

[1] https://huggingface.co/docs/transformers

set provided by the shared task organizers. The outcome of the NLI model can be binary: either entailment or contradiction. We use the metrics including baseline F1-score, faithfulness, and consistency, proposed by (Jullien et al., 2024) to calculate the performance of the models.

Among discriminative and generative models, we can observe that generative models including FlanT5 and SciFive outperform the discriminative models e.g. ClinicalBERT, BioLinkBERT, and BioClinicalBERT, in terms of faithfulness and consistency. Given the moderate amount of labeled data for this clinical inference task and textual data as input to the model, which is of low-dimensionality in the latent space compared to high-dimensional vision and speech data, this enables generative models to perform well by learning the joint probability distribution of the input features and the class labels. However, the DeBERTa-lg, which is a discriminative model, achieves the highest F1 scores among all discriminative and generative models. This is likely because the task involves simple binary classification, which can be comparatively easily performed by a discriminative model by separating the data points in the data manifold through a decision boundary. Therefore, for the clinical inference task with the provided dataset, both generative and discriminative models can be useful and demand empirical evaluation.

Table 3 also demonstrates that knowledge augmentation by adding evidence from all the sections does not improve the performance of the models in almost all cases. One possible reason is that adding more information makes it more challenging for the models to extract the relevant information. Also, by increasing the input length, the model struggles with high-dimensional input space.

Among the models, only DeBERTa-lg and FlanT5 are general-purpose models while the rest are tailored for the clinical domains by pre-training the models on domain-specific data. Also, DeBERTa-lg and SciFive are the only task-specific NLI models studied in this work. This is intriguing to observe that although DeBERTa-lg is not pre-trained on clinical data, it yields the highest F1-score. Thus, a model tailored to the task but not initially trained on domain-specific data may outperform a domain-specific model that lacks task specificity, demonstrating the importance of task-oriented adaptation rather than relying solely on domain-specific pre-training. This outcome contradicts our initial hypothesis that domain-specific pre-trained LLMs are necessary for superior performance.

Finally, the number of trainable parameters of the discriminative models is not found to be linked with model performance since discriminative BioLinkBERT, containing 340M parameters, performs either on par with or subpar than ClinicalBERT and BioClinicalBERT with 110M parameters each. However, the generative SciFive model, consisting of a larger number of trainable parameters than FlanT5, exhibits better performance in certain metrics e.g. faithfulness and consistency.

6 Conclusion

In this paper, we describe our system for the SemEval-2024 Task 2, dealing with NLI for clinical trials. Leveraging DeBERTa-lg, a discriminative pre-trained model tailored to the NLI task, we achieve a consistency score of 0.76, securing the 4th position out of 31 participants. Our exploration yields intriguing insights: both discriminative and generative models exhibit promise for this clinical inference task. In addition, we find that knowledge augmentation poses challenges for the model, possibly due to the higher dimensionality of the input space. Moreover, task-specific but not domain-specific models are found to be better performing than domain-specific but not task-specific models. However, it is worthwhile to mention that all the models are fine-tuned with the same clinical data. Interestingly, while the performance of the discriminative model is not affected by the number of parameters, it appears to influence the performance of generative models.

As part of future work, we intend to explore the applicability of parameter-efficient techniques including adapter-tuning (Houlsby et al., 2019), LoRA (Hu et al., 2021), etc. by deploying them for the clinical inference task.

7 Acknowledgments

The authors would like to thank the committee of SemEval2024, the organizers of Task 2, and the reviewers. Special thanks to Md Zobaer Hossain for providing continuous support.

References

Nancy E Avis, Kevin W Smith, Carol L Link, Gabriel N Hortobagyi, and Edgardo Rivera. 2006. Factors associated with participation in breast cancer treat-

ment clinical trials. *Journal of Clinical Oncology*, 24(12):1860–1867.

Hilda Bastian, Paul Glasziou, and Iain Chalmers. 2010. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9):e1000326.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Jay DeYoung, Eric Lehman, Benjamin Nye, Iain Marshall, and Byron C Wallace. 2020. Evidence inference 2.0: More data, better models. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 123–132.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Alistair Johnson, Tom Pollard, Lu Shen, Li-wei Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Celi, and Roger Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli. *Political Analysis*, 32(1):84–100.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. In *International conference on machine learning*, pages 6905–6916. PMLR.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Juraj Vladika and Florian Matthes. 2023. Sebis at SemEval-2023 task 7: A joint system for natural language inference and evidence retrieval from clinical trial reports. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1863–1870, Toronto, Canada. Association for Computational Linguistics.

Guangyu Wang, Xiaohong Liu, Zhen Ying, Guoxing Yang, Zhiwei Chen, Zhiwen Liu, Min Zhang, Hongmei Yan, Yuxing Lu, Yuanxu Gao, Kanmin Xue, Xiaoying Li, and Ying Chen. 2023. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. *Nature Medicine*, 29.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. In *Association for Computational Linguistics (ACL)*.

NLP at UC Santa Cruz at SemEval-2024 Task 5: Legal Answer Validation using Few-Shot Multi-Choice QA

Anish Pahilajani* and **Samyak R Jain*** and **Devasha Trivedi***
Baskin School of Engineering
University of California, Santa Cruz
{apahilaj,srajeshj,detrived}@ucsc.edu

Abstract

This paper presents our submission to the SemEval 2024 Task 5: The Legal Argument Reasoning Task in Civil Procedure. We present two approaches to solving the task of legal answer validation, given an introduction to the case, a question and an answer candidate. Firstly, we fine-tuned pre-trained BERT-based models and found that models trained on domain knowledge perform better. Secondly, we performed few-shot prompting on GPT models and found that reformulating the answer validation task to be a multiple-choice QA task remarkably improves the performance of the model. Our best submission is a BERT-based model that achieved the 7th place out of 20.

1 Introduction

The field of Natural Language Processing (NLP) has made significant strides in understanding and generating human language. Yet, specialized fields such as legal reasoning within the sphere of civil procedure pose distinct challenges. These challenges stem from the intricate nature of legal texts and the requisite domain-specific knowledge. In this paper, we present a solution to the problem posed in Semeval 2024 Task 5, which introduces a new NLP task and dataset focused on the U.S. civil procedure. Our approach in this task aims to evaluate the ability of large language models (LLMs) in interpreting and applying legal principles and laws to specific case questions. To support this paper, we have made our codebase publicly available as a GitHub repository.* This repository contains all our code and instructions how to run it. At the request of the task organizers, we have not included the dataset splits as they are meant to be private.

The dataset for this task is curated from "The Glannon Guide to Civil Procedure" (Glannon, 2019). It comprises a series of legal cases, each with a general introduction, a specific question related to U.S. civil procedure, and a possible answer candidate. For every answer choice, a comprehensive analysis is provided, rationalizing why it is correct or incorrect. A correct answer is labeled with 1, and an incorrect answer is labeled with 0. The dataset is available in English, and its training, validation, and test splits contain 666, 84, and 98 examples respectively.

2 Related Work

The domain of legal question-answering systems has witnessed substantial progress, utilizing cutting-edge computational techniques to address the complexities of legal discourses. A prime example of innovation in this field is the LEGAL-BERT system, introduced by Chalkidis et al. (2020), illustrating the enhanced efficacy of models tailored specifically for legal content through the pretraining of BERT (Devlin et al., 2019) on legal documents (Chalkidis et al., 2020). In addition, Khazaeli et al. (2021) achieved a notable breakthrough by developing a flexible legal question-answering system that goes beyond conventional query patterns, incorporating sparse vector search with a BERT-based re-ranking process. The expansion of legal corpora, notably with the Casehold corpus by Zheng et al. (2021), marked a significant stride forward. This work employed language models for legal analysis, setting a comprehensive standard for measuring model effectiveness in legal reasoning tasks using a dataset based on U.S. court cases. Furthermore, the creation of targeted NLP tasks has played a crucial role in the assessment of models and systems in this field. An important example is the task introduced by Bongard et al. (2022), which is the focus of this paper.

These works collectively underscore the diverse methodologies and technological advancements employed in legal question answering. Each of

*Equal contribution
*Code available here: https://github.com/devashat/UCSC-NLP-SemEval-2024-Task-5/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1309–1314

these contributions brings unique insights and solutions, paving the way for more sophisticated and efficient legal question-answering systems in the future.

3 System Overview

Our approach at creating a system entailed utilizing few-shot prompting on OpenAI's GPT-3.5 (OpenAI, 2023a) and GPT-4 (OpenAI, 2023b). We devised a script with a prompt, two examples of the expected model input and desired output, and used an altered version of the task dataset as our input queries for this system. We experimented with a few prompts for both GPT models, trying to figure out what gave us the best results. Following the guidance outlined in Bsharat et al. (2024) and White et al. (2023), we decided to structure our prompt in the following manner:

- A system instruction that describes to the model the structure of our data, the input it will receive, and what the model should return,
- Two examples from the training dataset, one with a correct answer prediction and one with an incorrect answer prediction,
- The dataset containing our questions and answer candidates.

Additionally, we also altered the dataset from a binary classification format to a multi-choice QA format. Rather than presenting individual question-answer pairs, each question was now accompanied by the entire set of potential answer choices. Figure 1 demonstrates a visual example of this restructuring. As can be inferred from Figure 1, the binary classification format of the dataset requires the system to label each answer choice as 0 or 1, whereas the multi-choice format requires the system to return one single answer prediction for each question. We chose to convert the dataset in this way to address an issue we found in the experimental phase of our work. This issue is elaborated upon in section 4.

A small note on our multi-choice QA format, we added an additional option "None of the Above" because in some cases, the training or the validation data had all incorrect answer choices listed for a given question. This was done to make sure that the model would not be forced to pick between multiple incorrect answers.

Binary Classification Format

```
Question:
<Question>

Context:
<Explanation>

Choice:
<Answer Candidate 1>

Question:
<Question>

Context:
<Explanation>

Choice:
<Answer Candidate 2>
```

Multi-Choice QA Format

```
Question:
<Question>

Context:
<Explanation>

Choices:
{0: <Answer Candidate 1>,
1: <Answer Candidate 2>,
2: <None of the Above>}
```

Figure 1: Difference between the original dataset format and our restructuring

Figure 2 shows our best performing system instructions for the binary classification format of the dataset and the multi-choice QA format of the dataset. We ran experiments on both dataset formats to compare system performance, and section 5 contains our results for evaluation metrics.

4 Experiments

4.1 Finetuning with BERT

To establish a solid baseline, we opted to fine-tune various BERT models. This process involved inputting both the question and its corresponding answer into the model, with the goal of generating an output label of either 0 or 1. We trained our model on the data for 500 epochs before having it predict. In the predictions we observed a propensity for both BERT models to disproportionately favor the 0 label, a phenomenon likely stemming from the dataset's natural imbalance due to it being formatted for binary classification. Since the source material for the dataset is in a multiple choice format, there are inherently more answers with the 0 label than answers with the 1 label, and a predic-

Multi-Choice System Instruction:

```
You are an AI legal expert with
expertise in U.S. Civil Procedure
and U.S. Civil Law, known for your
strong reasoning abilities. Your
task is to answer a Multiple
Choice Question in the legal domain.
Choose an answer only if you are
very confident, otherwise, select
"None of The Above."

You will be provided with:
1. question: A legal question
2. context: Additional context for
better understanding
3. choices: Multiple answer candidates

Your response should be a JSON with two
keys: "correct_answer" and "reasoning."
Place the correct answer exactly as
provided in the "correct_answer" key.
Provide a detailed explanation of your
reasoning in the "reasoning" key. Do
not add or remove any other text.

Your goal is to ensure accurate
answers and thorough reasoning.
```

Binary Classification System Instruction:

```
You are an AI legal expert with
expertise in U.S. Civil Procedure
and U.S. Civil Law, known for your
strong reasoning abilities. Your
task is to answer a question in
the legal domain.

You will be provided with:

1. question: A legal question
2. context: Additional context for
better understanding
3. answer candidate: an answer candidate
that can be either correct or incorrect

Your response should be a string with
length 1. You will be classifying a
correct answer as 1, and an
incorrect answer as 0.

Your goal is to ensure accurate
answers and thorough reasoning.
```

Figure 2: System Instructions for Both Dataset Formats

tive model would tend to prefer the majority label (Tanha et al., 2020). After noticing this issue, we tried experimenting with altering the dataset.

4.2 Data Augmentation

To address the challenge of our model's tendency to overfit on the 0 label, we explored incorporating the Casehold corpus into our dataset. Casehold (Zheng et al., 2021), a rich legal corpus derived from the Harvard case law collection and spanning from 1965 to the present, was initially formatted for multi-label use, offering a wealth of potential answers for each question. Despite our efforts to adapt this corpus into a binary format to align with the organizers' dataset format, we encountered persistent overfitting issues, leading us to believe that trying the balance the dataset would not yield any productive results.

Subsequently, we reverted to using solely the task dataset and refined our approach by integrating each question's text with its corresponding explanation to provide more context. This was done with the hope that our model would use the additional input to steer its prediction in the correct direction. However, this addition faced a technical bottleneck due to the 512-token limit inherent in the BERT models, prompting us to investigate alternative large language models (LLMs) that could handle the larger input size. We decided to explore two options. The first was finetuning Longformer because it uses windowed attention and can handle longer context lengths (Beltagy et al., 2020). The second was exploring few-shot prompting with GPT-3.5 and GPT-4 as we could run further experiments also comparing how the GPT models do with both formats of the dataset, if the overfitting issue would persist or if one dataset format would outperform the other. We also wanted to try few-shot prompting as Brown et al. (2020) found it to be a better approach for QA tasks than finetuning.

4.3 Finetuning Longformer

Finetuning Longformer helped us resolve the context length limit that we ran into with BERT, but it did not yield better results. Considering that in our experiments with BERT we found LEGAL-BERT to be the better performing model (see section 5), we decided to use a Longformer model with legal context embedded into it. This legal Longformer model is devised by Chalkidis et al. (2023) and is a derivative model of a base RoBERTa trained on the LexFiles corpus (Chalkidis et al., 2023).

Using this legal Longformer model, we were able to incorporate the explanation feature into our input. Our input was explanation, question, answer. We first ran the finetuning code for 100 epochs where we ran into the same problem of overfitting. In fact, our F1 score would not go above 44.37 on the validation set - the model would only predict 0 labels and performed worse than finetuning

the BERT models. We then increased the number of epochs to 500, but that also did not show an improvement in F1 scores on the validation set. We believe that Longformer performed worse than LEGAL-BERT due to differences in their pretraining corpora.

4.4 Few-Shot Prompting with GPT-3.5 and GPT-4

Our experimentation with GPT-3.5 and GPT-4, through few-shot prompting, offered promising directions. Notably, this method enabled us to effectively incorporate even the analysis feature of the dataset within the context limit, achieving an impressive F1 score of 90 on the validation set. Despite this success, the approach did not consistently extend to the test set, suggesting that using analysis to predict correct answers has its limitations, as the test set inputs lacked the feature.

In our final strategy to mitigate the dataset's imbalance, we shifted from binary to multi-choice classification, allowing for a more nuanced model assessment. This change meant our model now aimed to identify the correct answer from a set of options, rather than simply labeling each answer as 0 or 1. Reapplying few-shot prompting to GPT-3.5 and GPT-4, with the dataset's adjusted format, led to our most improved performance on the dataset.

For the prompting experiments, we used the OpenAI API. We ran our prompting code for 3 epochs, and did not alter any other hyperparameters.

4.5 Rule-based Algorithm Application

After successfully implementing a prediction system, we increased our F1 score and accuracy by applying a rule-based algorithm tailored to the characteristics of each dataset. Recognizing the inherent imbalance within the datasets, we devised a strategy where if all answers to a question were labeled as 0 in the training and validation sets, then the answer for said question in the test set was presumed to be labeled as 1. Conversely, if there were any correct answers in the training or validation sets, the data entries with the corresponding question were considered incorrect in the test set. This adjustment allowed us to enhance our performance metrics significantly for the baseline BERT models and the GPT-3.5 and GPT-4 predictions, giving us the metrics outlined in Table 4 and Table 5 for the test dataset. We only utilized this technique for the competition part of the task, as we wanted to see how high we could score. We did not submit our predictions with the GPT models.

5 Results

Our submission to the SemEval task ranked 7th out of 20 on the competition leaderboard. What we submitted to the task competition was our best performing finetuned BERT model after applying the rule-based algorithm. This was not our best method, as we were able to achieve higher metrics through subsequent experimentation. Our best results overall can be seen in Table 5, which stem from few-shot prompting on GPT models using the multi-choice QA format of the dataset, and then applying the rule-based algorithm to the predictions generated.

Table 1 presents our F1 score and accuracy across the two BERT models we chose to fine-tune. Our experiments not only aligned with but also surpassed the benchmarks established by the task organizers (Bongard et al., 2022), achieving a 0.2 increase in F1 score by merely utilizing the question and answer features in the input coupled with our fine-tuning approach. When comparing the baseline results from Bongard et al. (2022) with our own, we found that our question, answer input alone had a similar score to their input that also utilized the explanation feature. They achieved a 65.73 F1 score, whereas our submission to the task competition achieved a 65.99 F1 score as shown in Table 4.

Our best results without utilizing the rule based algorithm came from few-shot prompting with GPT models using the multi-choice QA format of the dataset. Table 2 shows these result metrics, while Table 3 shows the metrics of few-shot prompting using the binary classification format of the dataset for comparison.

Model	F1 Score	Accuracy
BERT	57.56	**73.47**
LegalBERT	**63.27**	72.45

Table 1: Baseline fine-tuned BERT models and their performance on test set.

6 Conclusion

Our investigation into the application of Large Language Models (LLMs) in the domain of legal reasoning for civil procedure, as a contribution to Se-

Model	Test F1 Score	Test Accuracy
GPT-4	**71.70**	**80.61**
GPT-3.5	62.21	72.45

Table 2: Results of Multi-Choice QA Few Shot Prompting on GPT Models.

Model	Test F1 Score	Test Accuracy
GPT-4	**68.77**	**73.47**
GPT-3.5	48.64	48.98

Table 3: Results of Binary Classification Few Shot Prompting on GPT Models.

Model	F1 Score	Accuracy
BERT	59.99	74.49
LegalBERT	**65.99**	**74.49**

Table 4: Results of combining a rule-based algorithm with finetuning on BERT models.

Model	Test F1 Score	Test Accuracy
GPT-4	**74.68**	**82.65**
GPT-3.5	64.13	73.47

Table 5: Results of combining a rule-based algorithm with multi-choice few shot prompting on GPT models

mEval 2024 Task 5, has led us to several significant insights. These insights not only highlight the capabilities and limitations of current AI technologies in legal applications but also chart a course for future advancements in this intriguing intersection of technology and jurisprudence.

6.1 Best system

Our research identified that the application of multi-choice QA few-shot prompting on GPT-4 was the most effective method, achieving an F1 score of 71.70 and an accuracy of 80.61 on the test dataset. A significant insight from our experiments is the inherent limitation encountered with BERT models, notably their 512-token context length constraint. This limitation poses a unique challenge in legal reasoning tasks, where the richness and complexity of legal texts often necessitate a comprehensive contextual understanding that exceeds the input capacity of traditional models. By successfully navigating these constraints with GPT-4's advanced capabilities, our approach demonstrates the benefits of leveraging the more flexible and expansive context handling offered by newer generation models to effectively process and interpret dense legal information.

6.2 Impact of analysis feature

The inclusion of an analysis feature significantly improved LLM performance during the fine-tuning process on both the training and validation datasets. However, the anticipated benefits of this feature did not extend to the test dataset, likely due to differences in input structure between the training/validation and test phases. This suggests a potential overfitting problem, indicating that while models may become adept at recognizing patterns in training data, they may not necessarily understand the fundamental legal reasoning principles underlying the data.

6.3 Format of Dataset

The imbalance of the dataset, coupled with the fact that it was primarily sourced from a single textbook, introduced a challenge in preventing models from exploiting its predictable structure. To foster more rigorous and analytically profound datasets in this research domain, we propose diversifying the sources of dataset content. Additionally, we suggest that future datasets should challenge models to not only select the correct answer but also to generate the reasoning behind their choices. This method could provide a use case for the analysis feature, promoting a deeper understanding and application of legal principles, leveraging the full potential of Generative AI in legal reasoning.

6.4 Future Work

Exploring the integration of specific laws or precedents as a form of analysis presents an intriguing direction for enhancing the capabilities of Large Language Models (LLMs) in legal reasoning tasks. This approach deviates from the current format of analysis; explaining why an answer choice is correct or incorrect. Instead, it involves presenting the LLM with the relevant legal principles or statutes directly related to the question input. The model is then tasked with interpreting these legal documents to deduce the correct answer based on the law's stipulations.

Such a methodology could foster the model reaching a deeper level of engagement with the ma-

terial, as it not only challenges the model to grasp the nuances of legal language but also might help improve the model's ability to generalize from the principles of law to the specifics of individual cases, potentially leading to more accurate and legally sound predictions. As such, future work could involve curating or enhancing existing datasets to include these legal references, alongside developing model architectures and training methodologies that are adept at handling such complex, text-based inputs.

Acknowledgements We would like to express our gratitude to Professor Ian Lane, Professor Jeffrey Flanigan, Nilay Patel, and Jeshwanth Bheemanpally from University of California, Santa Cruz. Their comments, insights, and feedback helped us along the process of participating in this task and writing this paper.

References

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The Long-Document Transformer.

Bongard, L., Held, L., and Habernal, I. (2022). The Legal Argument Reasoning Task in Civil Procedure.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners.

Bsharat, S. M., Myrzakhan, A., and Shen, Z. (2024). Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2020). LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Chalkidis, I., Garneau, N., Goanta, C., Katz, D., and Søgaard, A. (2023). LeXFfiles and LegalLAMA: Facilitating English Multinational Legal Language Model Development. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15513–15535, Toronto, Canada. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Glannon, J. W. (2019). *The Glannon Guide to Civil Procedure*. Wolters Kluwer, New York, NY, 4 edition.

Khazaeli, S., Punuru, J., Morris, C., Sharma, S., Staub, B., Cole, M., Chiu-Webster, S., and Sakalley, D. (2021). A Free Format Legal Question Answering System. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 107–113, Punta Cana, Dominican Republic. Association for Computational Linguistics.

OpenAI (2023a). GPT-3.5 Model Documentation. `https://platform.openai.com/docs/models/gpt-3-5-turbo`. Accessed: 2024-02-05.

OpenAI (2023b). GPT-4 Model Documentation. `https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo`. Accessed: 2024-02-05.

Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., and Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, 7:1–47.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT.

Zheng, L., Guha, N., Anderson, B. R., Henderson, P., and Ho, D. E. (2021). When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset.

914isthebest at SemEval-2024 Task 4: CoT-based Data Augmentation Strategy for Persuasion Techniques Detection

Dailin Li[1], Chuhan Wang[1], Xin Zou [1], Junlong Wang[2], Peng Chen[1]
Jian Wang [1†],Liang Yang[1],Hongfei Lin[1]
[1]School of Computer Science and Technology, Dalian University of Technology, China
[2]School of Software, Dalian University of Technology, China
{ldlbest,wangchuhan,zouxin,jlwang,pengchen}@mail.dlut.edu.cn
{wangjian,liang,hflin}@dlut.edu.cn

Abstract

Memes are commonly used in online disinformation campaigns, particularly on social media platforms. They are primarily effective on social media platforms since they can easily reach many users. Semeval2024-Task4(Dimitrov et al., 2024), "Multilingual detection of persuasion techniques in memes", focuses on detecting persuasive methods across four languages: English, Bulgarian, North Macedonian and Arabic. Subtask 1 aims to identify the given text fragments of memes and which of the 20 persuasion techniques it uses, organized in a hierarchy. For the difficulty of this task and the fundamental role of text in the artificial intelligence area, we concentrate solely on this task. We develop a system using CoT-based data augmentation methods,in-domain pretraining and ensemble strategy that combines the strengths of both RoBERTa and DeBERTa models. Our solution achieved the top ranking among **33** teams in the English track during the official assessments. We also analyze the impact of architectural decisions, data construction and training strategies. We release our code at https://github.com/ldlbest/semeval2024-task4

1 Introduction

In the present digital era, persuasive communication is pivotal across diverse arenas, from political discourses to the viral spread of content on social media platforms. A nuanced comprehension of the intricacies of persuasion is indispensable in safeguarding against misinformation, upholding the integrity of information, and nurturing a constructive digital discourse.

In online communication, memes have become decisive for disseminating information and influencing opinions. The focus of this task centres on addressing the intricate task of identifying persuasive techniques within the textual content of memes. This paper addresses the "Textual Persuasion Technique Identification" task, emphasizing recognizing persuasive techniques within meme text. Our approach aims to deliver a robust multi-label classification system tailored to navigate the intricate challenges posed by this task.

We employ the Transformer (Vaswani et al., 2017) architecture. We introduce ensemble learning (Breiman, 1996), integrating one DeBERTa (He et al., 2021) model and four RoBERTa (Liu et al., 2019) models, each trained with different random seeds. In the pretraining phase of our system development, we utilize the in-domain pretraining method to improve our model's context and semantic comprehension. To bolster our dataset, we incorporate additional data from similar past tasks. Furthermore, we implemented the data augmentation technique, enhancing data diversity by employing data augmentation techniques.

Below is a summary of our contributions:

- We augment the training dataset with a Chain-of-Thought (CoT) based data augmentation method and improve our model's performance.
- Our system utilizes in-domain pretraining to enhance performance and leverages ensemble learning to combine DeBERTa and RoBERTa for further improvements.
- In task 1, we achieve first place on the English test set among **33** participants with an F1 score of **0.752**.

2 Background

2.1 Persuasion Techniques

Persuasive communication wields a critical influence across various sectors, including political rhetoric and the spread of online content. Such communication is instrumental in guiding public discourse and moulding opinions, ensuring its significance in the modern digital landscape (Yu et al.,

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1315–1321

2021). This task extends these concepts to analyzing memes, an increasingly prevalent medium on social media and internet platforms. With their distinctive blend of fun and brevity, memes deftly navigate the web to share insights, provoke conversation, and distribute knowledge.

In our task, we concentrate on identifying persuasive techniques within textual content. According to the work of Piskorski et al. (2023), this task involves categorizing textual persuasive techniques into three subtypes: ethos, pathos, and logos. This taxonomy is further amplified to include 20 subordinate precise methods, providing an extensive framework for understanding and interpreting the art of persuasion in digital content.

2.2 Data Augmentation

Data Augmentation (DA) techniques are usually initially explored in computer vision (CV), but they have been relatively slow to gain traction in NLP. Challenges arise due to the discrete nature of language, which rules out continuous noise and makes it hard to maintain diversity (Feng et al., 2021). Although challenges exist, the evolution of NLP has led to an increasing demand for exploring tasks and domains with insufficient training data. Consequently, this trend has resulted in the proliferation of research studies utilizing DA techniques. One classical DA method is back translation (Sennrich et al., 2016), which involves translating the text into another language and then back into the original language. Wei and Zou (2019) proposed EDA to improve the performance of text classification tasks and exhibit solid results on smaller datasets. These techniques are helpful for augmenting data, but they only modify the original text in fundamental ways, sometimes even changing the entire meaning of the sentence. Additionally, Chen et al. (2023) proposed knowledge-guided data augment based on the semantic relations of the knowledge graph.

Recently, Large Language Models (LLMs) can provide a unified solution for various NLP tasks and achieve competitive performance (Zhao et al., 2023). For example, GPT-3 (Brown et al., 2020) and ChatGPT (Ouyang et al., 2022) have demonstrated strong performance in various NLP tasks and benchmark tests (Qin et al., 2023). Furthermore, LLMs play a role in data augmentation, enhancing their utility in multiple applications. Dai et al. (2023) introduced a text data augmentation approach based on ChatGPT, which can be used in downstream model training. Abaskohi et al. (2023) proposed Contrastive Paraphrasing-guided Prompt-based Fine-tuning of Language Models (LM-CPPF). To enhance the capacity of LLMs for intricate reasoning tasks, Wei et al. (2022) proposed Chain-of-Thought (CoT). Inspired by the effectiveness of the CoT method, we leverage CoT prompts to generate paraphrases used for data augmentation, ensuring the preservation of semantic consistency while significantly expanding our dataset. This augmentation strategy contributed to the enhanced performance of our model.

3 Our System

As depicted in Figure 1, our system comprises the following parts: Data Model, in-domain Pretraining, RoBERTa encoder, DeBERTa encoder and Soft Voting. The final prediction is obtained as $\hat{y}$. We ignore the hierarchical structure of the labels and define it as a multi-label classification problem (Tsoumakas and Katakis, 2007) for the labels of the training dataset are all final nodes of the graph.

3.1 Dataset Construction

We construct different data augmentation datasets based on various data augmentation strategies. In practice, while efforts to balance label distribution (such as using techniques like Nlpaug and CoT) aim to increase the number of samples for less frequent labels, it is essential to note that, since the data typically involves multiple labels, they can also result in the expansion of more frequent labels.

Nlpaug: We identify labels corresponding to train data with fewer than 1000 entries. Subsequently, we employ the nlpaug (Ma, 2019) library for these data points to implement data augmentation. Specifically, we utilize the method of synonym replacement, generating new training samples by substituting words in the text with their synonyms. This approach enhances the diversity of training data, thereby enhancing the model's robustness to different text inputs. Using this method, we augment more than 5700 data entries in total.

CoT-based Paraphrasing-Guided Data Augmentation: We filter data corresponding to labels that occupy less than 0.16 of the entire label distribution and rewrote these entries using GPT-3.5, generating 10,000 entries through this method. The LLM can fully understand the context and focus on improving the targeted content by explaining the labels and tasks and providing a specific description

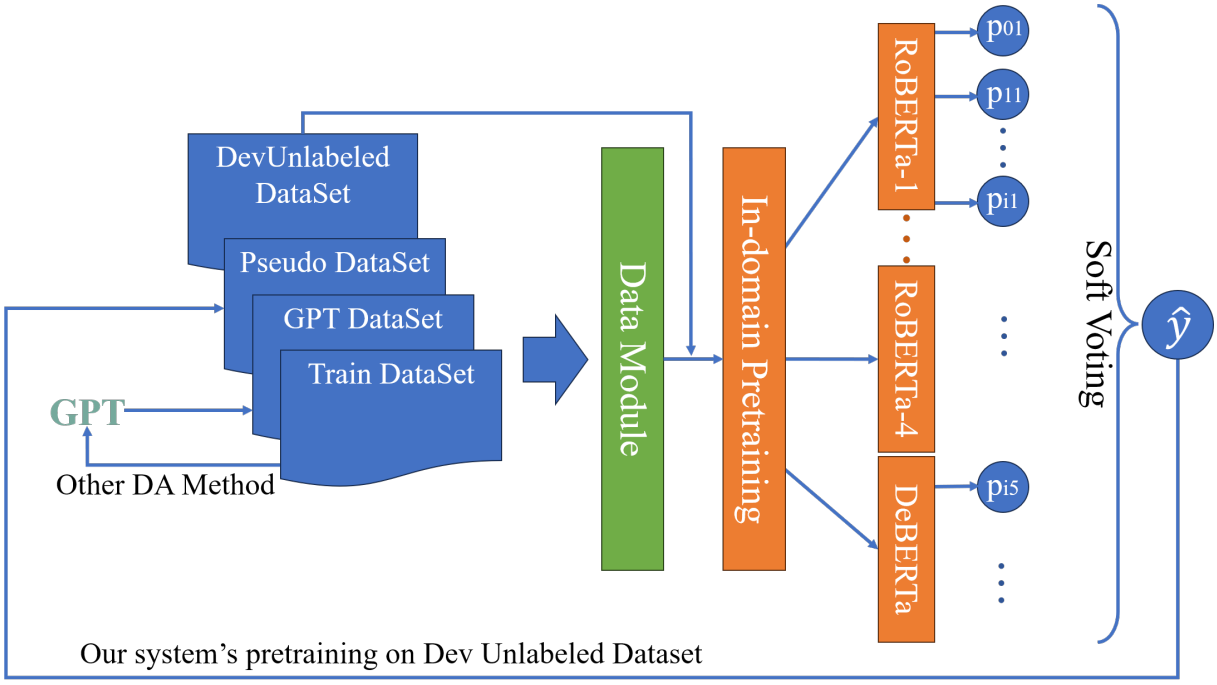


Figure 1: The overall architecture of our system.

of the problem and data that need to be rewritten. Applying the CoT technique enables the model to acquire more information and generate improved augmented data.

Input

User : Label Flag-waving means …
I will send you data with labels Flag-waving, Thought-terminating cliché at the same time, please elaborate again.
AI : Certainly! Feel free to provide the data with labels Flag-waving, Thought-terminating cliché and I'll do my best to help you analyze and elaborate on them.
User : Generate a paraphrase of the following text that uses different words and sentence structures while still conveying the same meaning. **REAL PATRIOTISM IS A WILLINGNESS TO CHALLENGE THE GOVERNMENT WHEN IT'S WRONG.**

Output

AI : True loyalty to one's country is demonstrated through a readiness to question the government's actions when they are unjust.

Figure 2: An example of using CoT for data augmentation.

We illustrate Figure 2. In the third round of our conversation with GPT-3.5, we use the instruction "Generate a paraphrase of the following text using different words and sentence structures while still conveying the same meaning" because it accurately describes the task with its instructions. Abaskohi et al. (2023) proved its effectiveness as an instruction template.

Pseudo-labeling: We use our model to classify 1000 data points on the dev dataset and 1500 on the test dataset. Pseudo-labelling (Lee, 2013) employs labelled data for training and utilizes information from unlabeled data to enhance the model's performance. The objective is to make more complete use of available data resources and improve the model's performance.

3.2 In-domain Pretraining

We utilize Masked Language Model (MLM) pertaining to all data, including data from SemEval 2023 task 3, which injects in-domain knowledge of our training datasets, thereby encouraging better learning outcomes for the model.

For a given input text x, we first tokenize it to obtain the tokenized representation $x_{tokenized}$ and truncate or pad them according to the maximum sequence length. For each text, a certain proportion of tokens are randomly masked based on the MLM probability and replaced with the masked token $[MASK]$. We use cross-entropy loss as the loss function for the masked language model. We compare the model's predicted probabilities for each token position with the actual token's one-hot encoding and calculate the cross-entropy loss.

Method	Recall	Precision	F1-score
BCAmirs	0.732	0.668	0.699
OtterlyObsessedWithSemantics	0.755	0.648	0.697
TUMnlp	0.714	0.638	0.674
GreyBox	0.688	0.652	0.670
BCAmirs	0.690	0.640	0.664
LomonosovMSU	0.632	0.674	0.652
NLPNCHU	0.706	0.604	0.651
Baseline	0.300	0.477	0.369
Our System	**0.836**	**0.684**	**0.752**

Table 1: Comparison of the performance between other team's models on Task 1 English test dataset.

$$L(t, \hat{t}) = -\frac{1}{N}\sum_{i=1}^{N} t_i \log(\hat{t}_i) \quad (1)$$

where t is the encoding of the true token, and $\hat{t}$ is the probability distribution of the model's predictions.

3.3 Ensemble Learning

We train four RoBERTa models and one DeBERTa model using different random seeds. We integrate them using the soft voting approach, which averages the predicted probabilities of each label from all five models. Given predictions $p_{i1}, ..., p_{iN}$ for class i from these models, we employ the following formula to obtain the final prediction $\hat{p_i}$

$$\hat{p_i} = \frac{1}{N}\sum_{j=1}^{N} p_{ij} \quad (2)$$

We then set a threshold of 0.25, where $\hat{p_i}$ greater than the threshold is chosen as the predicted label.

3.4 Low-Resource Languages

Since our training data is limited to English, we utilize GPT-3.5 to translate Bulgarian, North Macedonian and Arabic datasets into English. Subsequently, we perform inference on the translated data. The results obtained from these experiments can be found in the A.1. For the loss of information during translation, our system gets a relatively low F1 score in these languages.

4 Experimental Setup

The completion is based on PyTorch, Transformers and Pytorch-Lighting. During training, we set the batch size as 16, the learning rate as 3e-5, and the warmup steps ratio as 0.3. Five seeds(42,3407,114514,4096,1234) are used for the label ensemble. We use the AdamW optimizer and the cosine decay scheduler with a power of 0.01. We set a maximum epoch of 7. All experiments are run on one RTX 4090 GPU.

We create three additional datasets for the experiment: GPTDataset, PseudoDataset, and GPT-PseudoDataset (GPT-PDataset). The GPTDataset contains 7,500 training data and 10,686 sentences generated by LLM. PseudoDataset contains 7500 training data and test dev dataset labelled by the Ensemble model. GPT-PDataset is a union of GPTDataset and PseudoDataset.

5 Result and Analysis

In this section, we display our results and analyze the impact of each component through ablation studies.

5.1 Results

In this competition with 33 teams, we achieve first place with a hierarchical F1 score of 0.75247. We outperform the official baseline by 0.38382. The result is shown in Table 1.

We conduct a comparative analysis between our system and other models, including LLMs on the Task 1 English dev set, revealing the superior performance of our approach.

As illustrated in Table 2, GPT-3.5 (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023) utilized zero-shot learning, where only label meanings were provided in textual form without specific examples. We compare ourselves with other participating teams; the results are shown in Table 2. We achieve fourth place with a Hierarchical F1 score of 0.67833. Our performance is significantly better than the official baseline by 0.32010.

Method	Recall	Precision	F1-score
GPT-3.5	0.457	0.385	0.418
GPT-4	0.432	0.482	0.456
CLaC	**0.967**	**0.808**	**0.881**
OtterlyObsessedWithSemantics	0.754	0.636	0.690
GreyBox	0.716	0.657	0.685
EURECOM	0.702	0.650	0.675
Baseline	0.291	0.466	0.358
Our System	0.727	0.636	0.678

Table 2: Comparison of the performance between LLM and other team models on Task 1 English dev.

5.2 Ablation Study

We also conduct ablation experiments to validate our designs, including the encoder model, data modules, training strategy and ensemble.

Encoder Model We build our baseline model with BaselineDataset and BCE loss and run experiments to find out the best encoder model among RoBERTa (Liu et al., 2019), BERT (Devlin et al., 2019) , DeBERTa (He et al., 2021) , etc. As shown in Table 3, the large version of DeBERTav3 achieves the best score. Due to limited computility, we chose RoBERTa as our base model.

Method	F1-score
$BERT_{base}$	0.542
$BERT_{large}$	0.576
$RoBERTa_{base}$	0.614
$RoBERTa_{large}$	0.632
$DeBERTav3_{large}$	**0.649**

Table 3: F1 score of different Transformer-based models.

Training strategy We apply the in-domain pretraining on all encoder-based models to facilitate their performance on the downstream task. The result is shown in Table 4. For the five models, the F1 score improved by 0.2.

Method	F1-score
$BERT_{base}$	0.599
$BERT_{large}$	0.613
$RoBERTa_{base}$	0.630
$RoBERTa_{large}$	0.664
$DeBERTav3_{large}$	**0.667**

Table 4: F1 score of models after MLM training.

Data Module We use the best encoder model based on the result of dev datasets for ablation experiments on different datasets, including PseudoDataset, GPTDataSet, and GPT-PseudoDataset. The results are shown in Table 5.

Method	dataModule	F1-score
$RoBERTa_{large}$	GPTDataSet	0.685
$DeBERTav3_{large}$	GPTDataSet	0.700
$RoBERTa_{large}$	PseudoDataset	0.707
$DeBERTav3_{large}$	PseudoDataset	0.704
$RoBERTa_{large}$	GPT-PDataset	0.718
$DeBERTav3_{large}$	GPT-PDataset	**0.719**

Table 5: results on different dataset.

Ensemble Our ensemble approach can significantly improve performance. We integrate the results of different seeds and models based on their performance on the dev set. The result is shown in A.2, where we can see our ensemble approach outperforms the best single model by 0.15 F1 score over the dev set.

6 Conclusion

This paper details the architecture and performance of our multi-label classification system designed for the Persuasion Techniques Detection task. Our system achieves the highest rank for English in the leaderboard, signalling a notable accomplishment in the competitive framework. A comprehensive analysis of the data characteristics and model dynamics informs the strategic modifications we institute to the dataset construction and model training strategy. The efficacy of these refinements is corroborated by extensive empirical evaluation.

For future research, exploring methods to integrate the informational richness of hierarchical labels within the multi-label classification framework and fully exploiting LLMs to identify persuasion

techniques remain promising avenues for further exploration.

References

Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. 2023. LM-CPPF: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 670–681, Toronto, Canada. Association for Computational Linguistics.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24(2):123–140.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Peng Chen, Jian Wang, Hongfei Lin, Di Zhao, Zhihao Yang, and Jonathan Wren. 2023. Few-shot biomedical named entity recognition via knowledge-guided instance generation and prompt contrastive learning. *Bioinformatics (Oxford, England)*, 39(8).

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. 2023. Chataug: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. A survey of data augmentation approaches for nlp. In *Findings*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Dong-Hyun Lee. 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

OpenAI. 2023. Gpt-4 technical report. Available at https://arxiv.org/abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

J. Piskorski, N. Stefanovitch, V-A Bausier, N. Faggiani, J. Linge, S. Kharazi, N. Nikolaidis, G. Teodori, B. De Longueville, B. Doherty, J. Gonin, C. Ignat, B. Kotseva, E. Mantica, L. Marcaletti, E. Rossi, A. Spadaro, M. Verile, G. Da San Martino, F. Alam, and P. Nakov. 2023. News categorization, framing and persuasion techniques: Annotation guidelines. Technical Report JRC132862, European Commission, Ispra.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *ArXiv*, abs/2302.06476.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China. Association for Computational Linguistics.

Seunghak Yu, Giovanni Da San Martino, Mitra Mohtarami, James Glass, and Preslav Nakov. 2021. Interpretable propaganda detection in news articles. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1597–1605, Held Online. INCOMA Ltd.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv*, abs/2303.18223.

A Appendix

A.1 Model Result on Multilingual Datasets

Method	Recall	Precision	F1-score
English	**0.836**	**0.684**	**0.752**
Bulgarian	0.450	0.477	0.463
North Macedonian	0.340	0.401	0.369
Arabic	0.436	0.285	0.345

Table 6: Performance Metrics on Multilingual Datasets

A.2 Ensemble Model Result

Method	seeds	F1-score
$RoBERTa_{large}$	42	0.698
$RoBERTa_{large}$	3407	0.697
$RoBERTa_{large}$	4096	0.694
$RoBERTa_{large}$	1234	0.695
$RoBERTa_{large}$	1145145	0.696
$RoBERTa_{large}$	42,3407,4096	0.709
$RoBERTa_{large}$	42,3407,114514	0.710
$RoBERTa_{large}$	3407,4096,114514	0.710
$RoBERTa_{large}$	42,4096,114514	0.712
$RoBERTa_{large}$	42,3407,4096,114514	0.713
$RoBERTa_{large}$	42,3407,4096,114514	**0.718**
$DeBERTav3_{large}$	42	

Table 7: Ensemble of different methods and seeds

HaRMoNEE at SemEval-2024 Task 6: Tuning-based Approaches to Hallucination Recognition

Timothy Obiso and **Jingxuan Tu** and **James Pustejovsky**
Department of Computer Science
Brandeis University
Waltham, Massachusetts
{timothyobiso, jxtu, jamesp}@brandeis.edu

Abstract

This paper presents the Hallucination Recognition Model for New Experiment Evaluation (HaRMoNEE) team's winning (#1) and #10 submissions for SemEval-2024 Task 6: Shared-task on Hallucinations and Related Observable Overgeneration Mistakes (SHROOM)'s two subtasks. This task challenged its participants to design systems to detect hallucinations in Large Language Model (LLM) outputs. Team HaRMoNEE proposes two architectures: (1) fine-tuning an off-the-shelf transformer-based model and (2) prompt tuning large-scale Large Language Models (LLMs). One submission from the fine-tuning approach outperformed all other submissions for the model-aware subtask; one submission from the prompt-tuning approach is the 10th-best submission on the leaderboard for the model-agnostic subtask. Our systems also include pre-processing, system-specific tuning, post-processing, and evaluation.

1 Introduction

The HaRMoNEE team proposes two architectures to use on the SHROOM (Mickus et al., 2024) task: transformer model fine-tuning and large-scale LLM prompt tuning. First, we pre-process the data. We identify two fields from each example to use in our models.

For the fine-tuning models, the fields are then formatted into a single string with a separator token. We use three different training strategies with SHROOM data to improve performance on the test sets. Finally, we run the model on the test set and post-process the data to get a score. For prompt-tuning, the prompt is constructed around the two selected fields. Using the validation datasets, we experiment with two models and two prompts. We select the best of both and finally evaluate the test sets.

This paper reports our results from these experiments submitted during the SHROOM task's evaluation phase. We discuss the data, the task, our methods, and the experiments we ran. In addition, we analyze and discuss our results and make proposals for future work in hallucination recognition. We make our code and best results publicly available.[1]

2 Related Work

Several approaches have been taken to scoring faithfulness and identifying hallucinations. Laban et al. (2022a) proposed to use Natural Language Inference (NLI) to detect inconsistency in summarization tasks. They applied NLI to sentence pairs and aggregated the scores on the document level to obtain a faithfulness score. Lattimer et al. (2023) adopted a similar method by chunking the whole document into smaller pieces. However, they prompted LLMs to generate their scores. TrueTeacher (Gekhman et al., 2023) leveraged LLMs to generate synthetic data that could augment models' ability to identify factual inconsistencies in summarization tasks. AlignScore (Zha et al., 2023) is a unified evaluation metric for factual inconsistency that is based on the information alignment between two arbitrary text pieces. SelfCheckGPT (Manakul et al., 2023) proposed using LLMs, particularly GPT models, to generate multiple potential consistent/contradictory responses as a task-agnostic method for hallucination detection and fact-checking.

Several datasets for NLI and hallucination are commonly used to pre-train and fine-tune models for these tasks. SNLI (Bowman et al., 2015) is a large NLI dataset collected from image captions. It consists of sentence pairs labeled as entailment, contradiction, or neutral. PAWS (Zhang et al., 2019) proposed a new dataset for paraphrase identification that features non-paraphrase pairs with high lexical overlap. Honovich et al. (2022) stud-

[1] https://github.com/brandeis-llc/shroom

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1322–1331

Field	Value
task	"DM"
src	"I'm an avid reader. What is the meaning of avid?"
hyp	"Having an intense interest in something."
tgt	"enthusiastic; keen; eager; showing great interest in something or desire to do something"
ref	"tgt"
model	"ltg/flan-t5-definition-en-base"
labels	["Not Hallucination", "Not Hallucination", "Hallucination", "Not Hallucination", "Not Hallucination"]
label	"Not Hallucination"
p(Hallucination)	0.2

Figure 1: Example data

ied the factual consistencies from text generation systems. They proposed TRUE, an automatic factual consistency assessment tool for tasks including summarization, dialogue generation, fact verification, and paraphrase detection. HaluEval (Li et al., 2023) proposed a dataset with human-annotated hallucinated instances specifically for evaluating the performance of LLMs in recognizing hallucinations.

Laban et al. (2022b) and Zha et al. (2023) both fine-tune pre-trained language models to obtain a score for each example indicating faithfulness between two texts or the likelihood of the presence of hallucination in one of the texts. Lattimer et al. (2023), Gekhman et al. (2023), and Manakul et al. (2023) all use LLMs in various ways to perform the same tasks.

3 Data and Task

SHROOM is split into two subtasks, model-aware and model-agnostic, with corresponding datasets. Task participants receive two sets of unlabeled training data, two sets of labeled validation data, and two unlabeled test sets. There is also a smaller labeled trial dataset.

3.1 Datasets

Table 1 shows a breakdown of each dataset. Table 2 shows a breakdown by label for the labeled datasets. The training sets are evenly split by example task. The validation and test sets are split 25:37.5:37.5. In each of the validation and test sets, there are more examples of "Not Hallucination" than "Hallucination".

Figure 1 is an example from the model-aware validation set. Every datapoint in the datasets includes the following fields: `task` (task: the task that the model is trained to perform, e.g. "DM", "MT", "PG"[2]), `src` (source: the text passed to the model), `hyp` (hypothesis: the model output), `tgt` (target: the "gold" text that the model should output), `ref` (reference: which field should serve as a reference for semantic information, e.g. "src" or "tgt").

The model-aware datasets include the additional field `model` (the name of the model used). The labeled data includes `labels` (the list of votes from all annotators), `label` ("Hallucination" or "Not Hallucination"), and `p(Hallucination)` (probability of hallucination: the likelihood that the model output contains hallucinated content).

The validation and test sets were labeled through crowdsourcing. Five annotators annotated each datapoint for the validation and test sets, and three annotators each for the trial set. The label of each datapoint is the label the majority of annotators chose. The probability of hallucination is reported as the ratio of "Hallucination" labels to all labels.

3.2 Task

This task is a binary classification task to determine whether the text generated by the LLM contains any hallucinated content. Accuracy is the main benchmark for this task. This task also uses Spearman's correlation coefficient, ρ, to measure the degree of agreement using `p(Hallucination)`.

From the example in Figure 1, we see the challenges of this task. Annotators were asked to determine whether the hypothesis, "Having an intense interest in something." contains hallucinated information. In addition, annotators know that the hypothesis was generated by the model `tg/flan-t5-definition-en-base`, a Definition Modeling model, from the input "I'm an avid reader. What is the meaning of avid?" Annotators are told to use "enthusiastic; keen; eager; showing great interest in something or desire to do something" as a semantic reference to make their

[2]DM = Definition Modeling, MT = Machine Translation, PG = Paraphrase Generation

Dataset	Trial	Train Agnostic	Train Aware	Val Agnostic	Val Aware	Test Agnostic	Test Aware
PG	9	10,000	10,000	125	125	375	375
MT	35	10,000	10,000	187	188	562	563
DM	36	10,000	10,000	187	188	563	562
Total	80	30,000	30,000	499	501	1,500	1,500

Table 1: Dataset Task Statistics, PG = Paraphrase Generation, MT = Machine Translation, DM = Definition Modelling

Dataset	Val AG	Val AW
Hallucination	218	206
Not Hallucination	281	295
Total	499	501
Dataset	**Test AG**	**Test AW**
Hallucination	611	551
Not Hallucination	889	949
Total	1,500	1,500

Table 2: Dataset Label Statistics, AG = model-agnostic, AW = model-aware

decision.

In this example from the model-aware validation dataset, four out of five annotators label this as "Not Hallucination". A simple probabilistic model may not be able to identify hallucinations. In this example, the target includes a few definitions of avid. Human annotators recognize that only matching one definition indicates no hallucination is present, while a probabilistic model will only see that very few tokens between these fields match. Here, one annotator did believe there was hallucination present, possibly because the hypothesis was not equally diverse or not semantically similar enough.

Another challenge is approximating the diversity of what humans consider a "Hallucination" to be across the three example tasks (PG, MT, DM). With all of this in mind, we formulate methods to identify hallucinations in LLM output.

4 Methods

4.1 Model Fine-Tuning

We first explore how transformer-based models would perform on this task. Due to the similarity of this task to NLI, models made for hallucination detection and NLI models are considered. For our preliminary experimentation, all models are trained and tested using only the validation datasets as they are labeled. Each of the validation datasets was split 80/20 into a train and test subset. The best-performing model is used for evaluation on the test set. Our preliminary experimentation shows that hallucination recognition models significantly and consistently outperform NLI models.

Despite the similarity of NLI to this task, the SHROOM dataset is more diverse than NLI datasets. SHROOM's three example tasks include many different forms of hallucinations that NLI models do not encounter as frequently or at all in their training. Models for hallucination recognition are usually trained on NLI datasets as well as others. This data diversity makes hallucination models especially well suited for the hallucinations and data in the SHROOM task.

After identifying the best model, we experiment with different fine-tuning approaches. We vary the number of epochs, which data the model is fine-tuned on, and the order of fine-tuning. Additionally, we inference these models before any training to serve as a baseline.

All models we fine-tune take in two texts as input. The first text is the frame of reference to determine if hallucinated material is present. The second text is the text that may or may not have a hallucination. The output of the model is a number from 0 to 1. A score of less than 0.5 indicates that a hallucination is "likely" present.

A score of 0 represents high confidence that hallucination is present; a score of 1 represents high confidence that hallucination is not present. Because this scale is inverse to the scale used by the task organizers, where `p(Hallucination)` being 1 indicates all annotators have chosen the label "Hallucination", the model output is subtracted from 1 and the difference is used as `p(Hallucination)`.

For our fine-tuning model architecture (as shown in Figure 3), we pre-process each datapoint by selecting two text fields to pass to the model and inverting the numerical scale to match the model output. After training, we post-process the output by re-inverting the numerical scale to get our `label` and `p(Hallucination)` fields.

Definition modeling is a task to generate a definition for a given word in context. In the example shown below, The source corresponds to the context; The target is the correct definition for this context; the hypothesis is the predicated definition from the model.

Example:
source: The sides of the casket were covered with heavy black broadcloth, with velvet caps, presenting a deep contrast to the rich surmountings. What is the meaning of surmounting?
target: A decorative feature that sits on top of something.
hypothesis: A sloping top.

Your task is to answer whether the hypothesis from the example contains any hallucination (e.g., incorrect semantic information unsupported or inconsistent with the source) and explain why. The target is inferred from the source without any hallucination. You should consider both the source and the target before making a judgment on the hypothesis.

The output should be formatted as a JSON instance that conforms to the JSON schema below.
As an example, for the schema {"properties": {"foo": {"title": "Foo", "description": "a list of strings", "type": "array", "items": {"type": "string"}}}, "required": ["foo"]}
the object {"foo": ["bar", "baz"]} is a well-formatted instance of the schema. The object {"properties": {"foo": ["bar", "baz"]}} is not well-formatted.

Here is the output schema:
{"properties": {"answer": {"title": "Answer", "description": "answer 'Yes' if the hypothesis contains hallucination; answer 'No' if the hypothesis does not contain hallucination", "type": "string"}, "reason": {"title": "Reason", "description": "a brief explanation to your answer", "type": "string"}}, "required": ["answer", "reason"]}

Figure 2: GPT Prompt 1 (Three Fields)

4.2 LLM Prompt-Tuning

Our team also uses LLMs as black-box hallucination detection systems. We first experiment with zero-shot classification on the validation sets. We use GPT-3.5 (Brown et al., 2020) and GPT-4 (OpenAI, 2023) with the prompt shown in Figure 2. If a model performs well on the validation sets, it would also be used to evaluate the test sets.

After choosing a model, we tune the prompt. The prompts we experiment with vary with respect to which fields are provided, the structure/order of the prompt, the inclusion of the task definition, and the overall verbosity of the prompt. One additional experiment we performed was the specific format of the model response. Asking a large-scale LLM to respond to a Yes/No question and give a reason may lead to many different responses. Although we expect the model to answer "Yes" or "No" when told to, it may ignore that part of the instruction and respond in a way that is correct but not directly interpretable by our post-processing such as "This contains a hallucination" instead of "Yes".

Asking the model to respond in JSON format increased the likelihood that the answer would be directly interpretable. Additionally, we ask our models to provide explanations for their "Yes" or "No" responses. While we do not use these to determine the `label` or `p(Hallucination)` for any datapoint, we found that asking LLMs to provide reasoning boosted performance. Shorter prompts such as those seen in Figure 5 were also used with these models. To ensure replicability, we set the hyperparameter `temperature` to 0.0. Setting this hyperparameter as such leads to increased determinism in responses.

Our post-processing assigns the `label` field "Hallucination" or "Not Hallucination" based on the value of the key `answer` found in GPT's JSON output. For these models, `p(Hallucination)` was set naively. Therefore, if the model returns `...{"answer": "Yes"}...`, `label` is "Hallucination" and `p(Hallucination)` is 1. When "Not Hallucination" is the label, `p(Hallucination)` is 0.

5 Experiments

Each datapoint contains up to three semantically relevant text fields, `hyp`, `src`, and `tgt`. In the model-agnostic subtask, these fields are always provided. In the model-aware subtask, `tgt` is left blank if the `task` is Paraphrase Generation (PG).

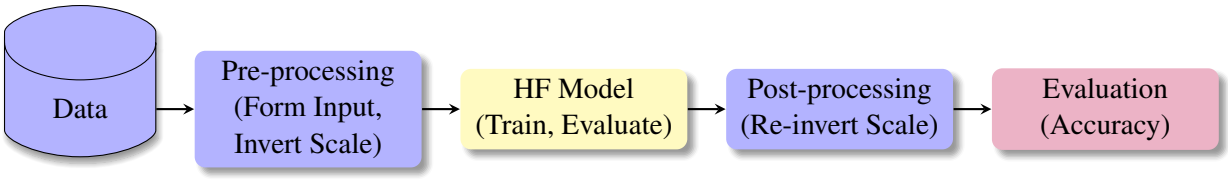


Figure 3: Fine-tuning architecture

Figure 4: Prompt-tuning architecture

Your task is to determine whether the hypothesis contains any hallucinations based on the target. (e.g., incorrect semantic information) and explain why. Only consider target and hypothesis when making the judgement. Your answer must start with 'Yes' or 'No'.

Example:
target: A decorative feature that sits on top of something .
hypothesis: A sloping top .

The output should be formatted as a JSON instance that conforms to the JSON schema below.
As an example, for the schema {"properties": {"foo": {"title": "Foo", "description": "a list of strings", "type": "array", "items": {"type": "string"}}}, "required": ["foo"]}
the object {"foo": ["bar", "baz"]} is a well-formatted instance of the schema. The object {"properties": {"foo": ["bar", "baz"]}} is not well-formatted.

Here is the output schema:
{"properties": {"answer": {"title": "Answer", "description": "answer 'Yes' if the hypothesis contains hallucination; answer 'No' if the hypothesis does not contain hallucination", "type": "string"}, "reason": {"title": "Reason", "description": "a brief explanation to your answer", "type": "string"}}, "required": ["answer", "reason"]}

Figure 5: GPT Prompt 2 (Two Fields)

We experiment by varying which of these fields get passed to our model and the structure of the model input.

We conduct a series of experiments using two approaches to detecting hallucinations. We first fine-tuned existing hallucination detection models using SHROOM validation data. Second, we experiment with a series of prompts to increase determinism and accuracy of LLMs on the same task. We show similarities of note between the best results of each architecture.

5.1 Fine-Tuning Experiments

The model that we find to perform the best on this task is a model to detect LLM-generated hallucinations. We find that the best results were obtained from this model when the input is of the form [CLS]+tgt+[SEP]+hyp for DM and MT. When tgt is not provided for the model-aware PG examples, the input [CLS]+src+[SEP]+hyp is used. This model is vectara/hallucination_evaluation_model on HuggingFace. This model took microsoft/deberta-v3-base (He et al., 2021) and trained it on two NLI datasets, SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018), as well as one paraphrase dataset, Paraphrase Adversaries from Word Scrambling (PAWS) (Zhang et al., 2019).

We first conducted our preliminary experiments on the split validation sets. After, we took the best-performing model, fine-tuned it on the entire validation set(s), and evaluated it on the test set(s). We experimented with varying the number of epochs, the datasets used, and the training order.

For both the model-agnostic and model-aware subtasks, we experimented with inferencing, 1-5

Dataset	Test AG		Test AW	
Metric	**acc**	ρ	**acc**	ρ
SHROOM Baseline	0.697	0.403	0.745	0.488
Corresponding Dataset	0.783	0.663	**0.813**	0.699
All Data	0.785	0.652	0.808	0.713
All + Corresponding	0.783	0.683	0.810	0.671
GPT-4 (Two Fields)	**0.814**	0.626	0.783	0.614

Table 3: Best results from each approach and the baseline results. Corresponding Dataset, All Data, and All + Corresponding are fine-tuning results; GPT-4 is prompt-tuning results

epochs of training on the corresponding validation dataset, 1-5 epochs of training on all validation data, and 1, 3, or 5 epochs of training on both validation datasets before one epoch of tuning on the corresponding validation dataset.

5.2 Prompt-Tuning Experiments

In addition to fine-tuning off-the-shelf models, we experiment with GPT-3.5 and GPT-4 using several prompts. After evaluating the validation sets on GPT-3.5 and GPT-4, further prompt tuning was done on GPT-4 due to its superior performance.

The first prompt we experiment with includes `hyp`, `src`, and `tgt`, task instructions, the example task definition, and the main prompt. The second prompt we used only included two fields (`hyp` and `tgt` for DM and MT, `hyp` and `src` for PG), task instructions, and the main prompt.

6 Results

Our best results from each training architecture are shown in Table 3. One notable similarity between our best results from each system is that they required pre-processing (data pruning). We obtain our best results by only including two of the three meaningful fields for each datapoint in both fine-tuning and prompt-tuning methods. For the fine-tuning methods, converting three fields into two via concatenation underperformed ignoring one field (`src` was ignored if `tgt` was provided). Additionally, removing the definitions of DM, MT, and PG from the prompt led to improved results.

Despite not making explicit use of the `model` field for the model-aware subtask, our models' best performances earned a higher spot in the model-aware subtask than the model-agnostic one. For the model-aware subtask, we use `src` when `tgt` is not provided for the model-aware PG examples. We also believe it to be due to the differences in other fields. For instance, the DM examples for the

Dataset	Test AG	Test AW
PG	0.789	0.875
MT	0.851	0.837
DM	0.794	0.747
All	0.814	0.813

Table 4: Accuracy of best submission to each subtask split by example task

# of epochs	0	1	2	3	4	5
Agnostic Data	0.756	0.769	0.776	0.776	0.783	0.783
All Data	0.756	0.779	0.777	0.777	**0.785**	0.780
All + Agnostic Data	0.756	0.783		0.775		0.783
Aware Data	0.794	0.804	0.804	0.805	**0.813**	0.803
All Data	0.794	0.796	0.808	0.808	0.808	0.797
All + Aware Data	0.794	0.808		0.810		0.795

Table 5: Table of accuracy data shown in Figure 6

model-aware and model-agnostic tasks are formatted differently. The model-aware examples ask for the definition explicitly at the end of the `src` field (e.g. "... What is the meaning of spoilage ?"). The model-agnostic examples put the word to define in tags (e.g. "The <define> sacrifice bunt </define> was fielded cleanly..."). These different tagging strategies reflect inputs different DM models take in. Based on our results in Table 4, it seems that our systems are better are recognizing hallucinations obtained via <define> tags.

Scores within .001 point of each other were obtained for each subtask using these systems, yet different systems performed the best for each. Our shorter prompt with GPT-4 produced our best results (#10) for the model-agnostic subtask, .117 points above the task organizer baseline. Training on the corresponding SHROOM validation dataset produced the best results (#1) for the model-aware subtask, .068 points above the task organizer baseline.

6.1 Fine-Tuning Results

Our team's first experiments on the test set involved varying the number of epochs, the training set(s), and the training order. These results for the model-agnostic and model-aware subtasks are shown in Figure 6 and Table 5. The 0 epoch results are from inferencing the model before training on SHROOM data. They serve as a baseline for all training strategies. This model's strong performance at inference for both test sets made it a strong contender for more fine-tuning. On the model-agnostic subtask, inferencing resulted in an accuracy of 0.756. On the model-aware subtask, it obtained a score of

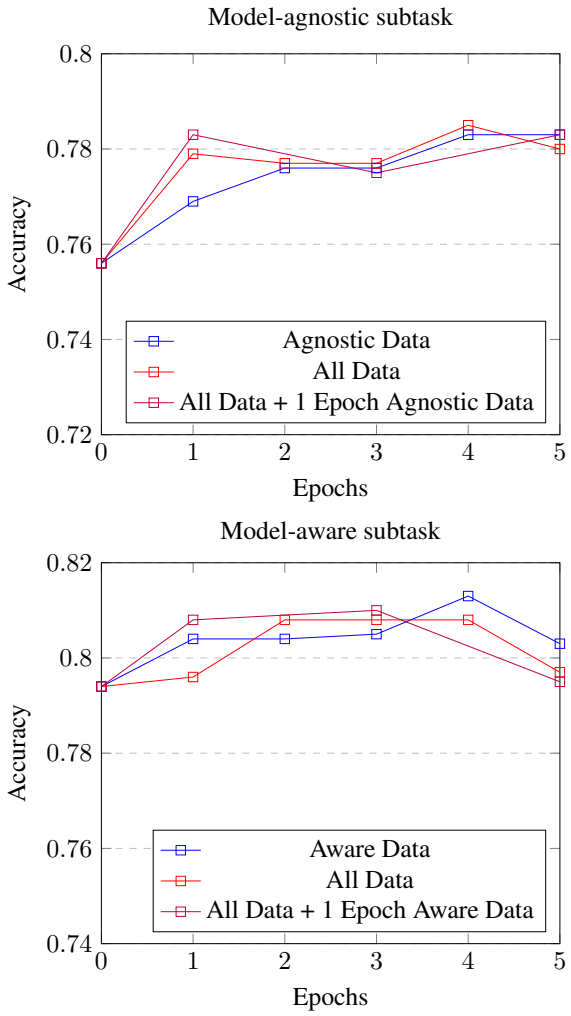


Figure 6: Line graphs showing accuracy on the test sets of both subtasks after fine-tuning

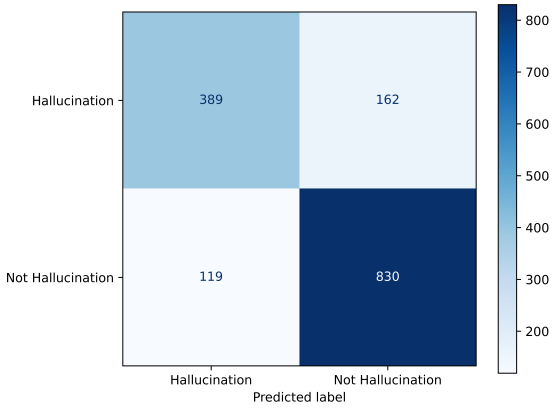


Figure 7: Confusion matrix for the #1 submission on the model-aware dataset (fine-tuning approach)

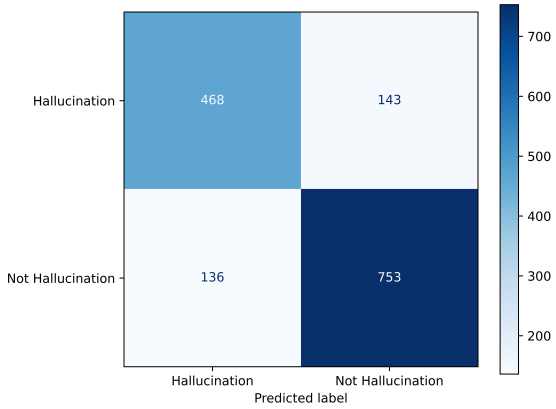


Figure 8: Confusion matrix for the #10 submission on the model-agnostic dataset (prompt-tuning approach)

Dataset	Val AG	Val AW
SHROOM Baseline	0.649	0.707
GPT-3.5	0.661	0.596
GPT-4 (Three Fields)	0.778	0.751
GPT-4 (Two Fields)	**0.782**	**0.773**

Table 6: Comparison of the baseline results to our results on the validation sets using accuracy

0.794.

Training with any SHROOM data led to better performance on the test set. For the model-agnostic subtask, our team's best result came from fine-tuning the model on the model-agnostic and model-aware datasets together for four epochs. This increased the accuracy from our inference baseline by 0.029 to 0.785. On the model-aware subtask, our team's best results were obtained after fine-tuning the model using only the model-aware data for four epochs. This increased the accuracy from the inference baseline by 0.019 to 0.813, our winning submission for the model-aware subtask.

As seen in Figure 7, this architecture performs well on the SHROOM model-aware subtask. Figure 9 and Table 4 show a breakdown by task. This model performed very well on PG and MT but much worse on DM. We believe this task has the lowest accuracy for the same reason we identified earlier in the paper. The target, source, and hypothesis fields may vary much more than in the other two tasks. The target may be much more or much less semantically rich than the hypothesis, which can be interpreted by human annotators and out models in many different ways. For the winning submission, the accuracy for the PG task was .875, MT was .837, and DM was .747.

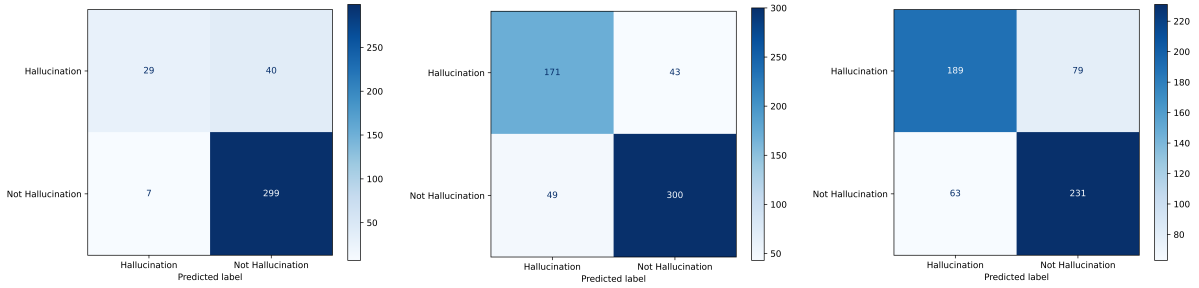


Figure 9: Confusion matrices for the #1 submission on the model-aware dataset (fine-tuning method) split by example task. Left to right: Paraphrase Generation, Machine Translation, Definition Modeling

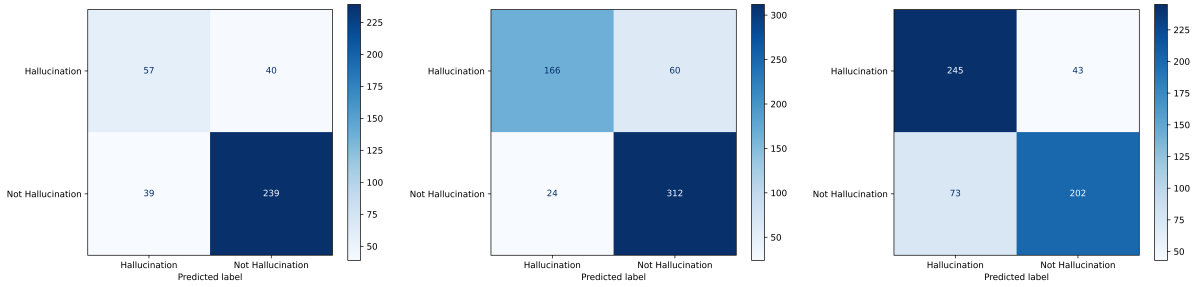


Figure 10: Confusion matrices the #10 submission on the model-agnostic dataset (prompt-tuning method) split by example task. Left to right: Paraphrase Generation, Machine Translation, Definition Modeling

6.2 LLM-Tuning Results

Our LLM experimentation results are shown in Table 6. Because we experimented with prompt-tuning on the validation sets, we were able to directly compare our results to the baselines provided by the task organizers. We found that GPT-3.5 performed worse than the baseline for one subtask and did not pursue further experiments with it. We tested two prompts on the validation sets.

Our first prompt (Figure 2) includes all meaningful fields that annotators had access to when labeling the data. For the model-agnostic subtask, this prompt outperformed the baseline by 0.129 with an accuracy of 0.778. it improved the accuracy in the model-aware subtask by 0.044 with an accuracy of 0.751.

Our second prompt (Figure 5) includes selected fields of meaningful information from each datapoint. It does not explain the example task (DM, MT, or PG) for the datapoint but still explains the shared-task instructions and output formatting instructions. This reduction in verbosity led to improved performance for both subtasks. For the model-agnostic validation set, this change resulted in a .133 point increase in accuracy above the baseline and a .004 point increase compared to the first prompt. For the model-aware validation set, this change resulted in a .066 point increase compared to the baseline and a .022 increase compared to the first prompt.

We also experimented with few-shot learning and found that both random examples and selected examples did not improve performance on either subtask. Overall, we found that a less verbose prompt outperformed a more verbose prompt indicating that GPT has difficulty making connections across large amounts of text. Additional information that GPT may not need in each prompt, such as the example task definition and the third field, adds noise to the prompt and impairs its ability to detect hallucinations.

The confusion matrices in Figures 8 and 10 show the performance of this architecture on the entire test set and on each task for the model-agnostic subtask. For this submission, the accuracy for the PG task was .789, significantly lower than the other example tasks. We believe this is because we used the `tgt` field for all tasks here as it is always provided. In the model-aware subtask, `tgt` is not provided for PG examples, so `src` is used instead. This may indicate that `tgt` is best for MT and DM, but `src` for PG even if `tgt` is provided. The accuracy for the MT task examples was .851, similar to the model-aware subtask. The accuracy for Definition Modeling was .794.

7 Discussion and Conclusion

In this paper, we present two systems for hallucination recognition, one transformer-based model fine-tuned on SHROOM data and one prompt-tuned zero-shot classification model using GPT-4. Our results show that both systems can better handle semantically complex tasks such as hallucination recognition when only semantically relevant information is provided. Pre-processing each example is essential to good performance on this task. From our results, fine-tuning using available labeled data from all tasks improves performance from the baseline. Additionally, pruning information such as over-explicit instructions, irrelevant fields, and definitions from prompts also improves performance from the baseline.

Some avenues we did not fully explore include training on pseudo-labeled training data, training on additional datasets besides SNLI, MultiNLI, and PAWS (specifically adversarial translation or word disambiguation datasets), as well as experimenting with dense paraphrasing and frame saturation methods as proposed by Tu et al. (2023). In this shared task, our team found that it is easier to harmonize by tuning fewer, clearer voices.

Acknowledgements

The authors would like to acknowledge the support and input from Kyeongmin Rim. This work was supported in part by NSF grant 2326985 to James Pustejovsky. The opinions and views reported herein are those of the authors alone.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023. TrueTeacher: Learning factual consistency evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2053–2070, Singapore. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022a. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022b. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Barrett Lattimer, Patrick CHen, Xinyuan Zhang, and Yi Yang. 2023. Fast and accurate factual inconsistency detection over long documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1691–1703, Singapore. Association for Computational Linguistics.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. ArXiv:2303.08896 [cs].

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task

on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Jingxuan Tu, Kyeongmin Rim, Eben Holderness, Bingyang Ye, and James Pustejovsky. 2023. Dense paraphrasing for textual enrichment. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 39–49, Nancy, France. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

VerbaNexAI Lab at SemEval-2024 Task 10: Emotion recognition and reasoning in mixed-coded conversations based on an NRC VAD approach

Santiago Garcia and **Elizabeth Martinez** and **Juan Cuadrado**
and **Juan Carlos Martinez-Santos** and **Edwin Puertas**
Universidad Tecnologica de Bolivar, Cartagena Colombia
epuerta@utb.edu.co

Abstract

This study introduces an innovative approach to emotion recognition and reasoning about emotional shifts in code-mixed conversations, leveraging the NRC VAD Lexicon and computational models such as Transformer and GRU. Our methodology systematically identifies and categorizes emotional triggers, employing Emotion Flip Reasoning (EFR) and Emotion Recognition in Conversation (ERC). Through experiments with the MELD and MaSaC datasets, we demonstrate the model's precision in accurately identifying emotional shift triggers and classifying emotions, evidenced by a significant improvement in accuracy as shown by an increase in the F1 score when including VAD analysis. These results underscore the importance of incorporating complex emotional dimensions into conversation analysis, paving new pathways for understanding emotional dynamics in code-mixed texts.

1 Introduction

Exploring emotion recognition in textual and multimodal conversations is crucial within Natural Language Processing (NLP) and Artificial Intelligence (AI). This domain addresses the complexity of human emotional expression, particularly challenged by the interlacing of multiple languages in code-mixed texts. Such code-switching, prevalent in digital communication, necessitates innovative computational strategies to decipher the embedded emotional substrates (Wang et al., 2022), presenting unique challenges for emotion recognition and understanding.

Recent advancements have highlighted the potential of complex neural architectures, like hierarchical transformers, to dissect the nuanced interplay between linguistic codes. This approach indicates a broader NLP trend that prioritizes models capable of parsing linguistic structures and decoding emotional cues within them (Cuadrado et al., 2023a). Significantly, the dynamic nature of conversational emotion and the phenomenon of emotion flips in multi-party interactions call for adaptive models that can trace these shifts accurately (Puertas et al., 2022).

Moreover, multimodal approaches that integrate visual, textual, and auditory cues are pivotal in capturing the essence of code-mixed interactions. These strategies convey sentiment and intention, underscoring the significance of non-verbal cues (Martinez et al., 2023). Additionally, the exploration of large language models for understanding complex conversational patterns has led to an evolving AI research landscape, where the efficacy of models like GPT in nuanced tasks such as sarcasm explanation and affect understanding in dialogues is rigorously evaluated (Cuadrado et al., 2023b).

Analyzing sociolinguistic features in digital social networks further enriches the discourse on digital communication's implications for emotion recognition and conversational AI. It includes bot detection, gender profiling, and community detection through sociolinguistic cues analysis (Moreno-Sandoval et al., 2019; Puertas et al., 2021, 2019). Moreover, the precision application of NLP methodologies, such as phonetic detection techniques for identifying hate speech spreaders on Twitter, showcases the necessity for targeted approaches to specific social media phenomena (Puertas and Martinez-Santos, 2021).

Our research aims to advance the understanding of NLP's multifaceted applications in digital interactions' integrity and authenticity. Through the analysis of polarity, emotion, and user statistics for fake profile detection, alongside multimodal emotion-cause pair extraction in conversations, we seek to significantly improve the comprehension of the complex interrelations between emotional expressions and their triggers (Moreno-Sandoval and Alvarado-Valencia, 2020; Wang et al., 2022).

In evaluating the incorporation of Valence,

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1332–1338

Arousal, and Dominance (VAD) scores from the NRC VAD Lexicon into our computational models, we observed a marginal performance improvement. Specifically, the inclusion of VAD scores resulted in F1 scores of 0.34 for Emotion Flip Reasoning (EFR) and 0.23 for Emotion Recognition in Conversation (ERC), compared to models without VAD scores, which achieved F1 scores of 0.32 and 0.20 for EFR and ERC respectively. These results underscore the nuanced challenges of accurately capturing emotional shifts in code-mixed conversations, paving the way for future research to refine and enhance emotion recognition systems in complex conversational contexts. Find here the GitHub repository[1]

2 Related Work

The recognition of emotions in code-mixed text and multimodal conversations has garnered increasing attention within the natural language processing (NLP) and artificial intelligence (AI) communities. The growing prevalence of code-switching in digital communication fuels this surge in interest and the multifaceted nature of human emotional expression.

Recent advancements in understanding code-mixed language semantics have underscored the potential of hierarchical transformer models to grasp the nuanced interplay between different linguistic codes (Sengupta et al., 2022). Through their ability to capture deep semantic relationships, these models offer a promising avenue for more accurate emotion recognition in code-mixed conversations. Such approaches align with the broader trend of employing sophisticated neural architectures to tackle the complexities of multilingual text processing.

Studies focusing on multiparty interactions have specifically addressed emotion flip in conversations, where the emotional trajectory can shift dramatically due to a single utterance or interaction (Kumar et al., 2023a, 2024a,b, 2022b). These studies highlight the dynamic nature of conversational emotion and the need for models that can adaptively reason about these shifts to maintain coherence and accuracy in emotion recognition tasks.

Multimodal approaches to sarcasm detection and humor classification in code-mixed conversations further illustrate the rich potential of integrating visual, textual, and auditory cues to enhance the understanding of conversational context and emotional undertones (Bedi et al., 2021). This multimodal perspective is critical in fully capturing the essence of code-mixed interactions, where non-verbal cues significantly convey sentiment and intention.

Exploring large language models' capability in logical reasoning and understanding complex conversational patterns points towards an evolving landscape in AI. Researchers have tested models like GPT (Generative Pre-trained Transformer) for their efficacy in nuanced tasks such as sarcasm explanation and effect understanding in dialogues (Xu et al., 2023). These inquiries into the logical capabilities of large models contribute to a deeper understanding of their potential applications in conversational AI and emotion analysis.

Moreover, research on explaining sarcastic utterances to enhance affected understanding in multimodal dialogues sheds light on the importance of context and the subtleties of human communication. Such work suggests that describing a particular emotional expression's underlying intent or cause beyond detecting sarcasm or emotion is crucial for advanced AI systems for naturalistic human-computer interaction (Kumar et al., 2023b).

The development of comprehensive datasets like MELD, which provides a multimodal multiparty dataset for emotion recognition in conversations, has been instrumental in advancing research in this area (Poria et al., 2018). These datasets not only facilitate the training and testing of sophisticated models but also enable the exploration of new methodologies for emotion recognition across diverse conversational settings.

As we move forward, the integration of insights from masked memory networks, transformer models, and intent-conditioned counter speech generation into the realm of emotion recognition and conversational AI promises to open new avenues for research and application (Poria et al., 2018; Kumar et al., 2022c; Christ et al., 2023). The collective efforts in these areas underscore the ongoing pursuit of more empathetic, contextually aware, and linguistically versatile AI systems capable of navigating the complexities of human emotion and communication.

3 Methodology

This section details the methodology adopted for analyzing emotion causes in multimodal conversations. Our approach, grounded in integrating

[1]https://github.com/VerbaNexAI/EmoVAD.git

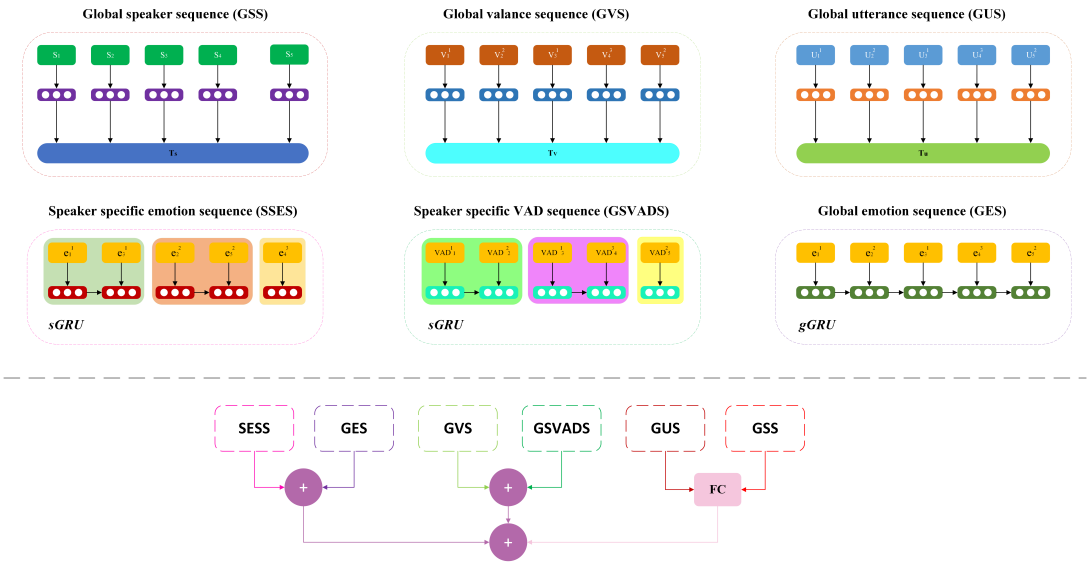


Figure 1: System General Pipeline

the NRC VAD Lexicon and computational models, aims to systematically identify and categorize emotion triggers. The methodology encompasses Emotion Flip Reasoning (EFR) for detecting shifts in conversation emotions and Emotion Recognition in Conversation (ERC), focusing on classifying these emotions accurately. By employing a combination of Transformer and GRU models, we enhance the analysis of valence, arousal, and dominance (VAD) scores, contributing to a nuanced understanding of emotional dynamics in conversations. All scripts and data related to this study are available at SemEval 2024 VerbaNex AI Repository.

3.1 Emotion Flip Reasoning

This section shows the proposed model and how this can identify the trigger for the corresponding emotional flip in the conversation; the way to identify it is by analyzing each utterance in the sequence; that is, the task is essential for a binary classification because we're trying to categorize each utterance responsible or not for the emotion flip. (Kumar et al., 2023a) propose the TGIF model, which contains the context of utterances, speakers, and emotions. This model explains how to process these three inputs through the pipeline. They propose four modules:

- **Global Utterance Sequence:** They use a Transformer (Vaswani et al., 2017) encoder architecture to push the $U = u_2, u_1, ..., u_i$ utterance distribution into a latent space, capture the global context of the dialogue
- **Global Emotion Sequence:** In this approach, we use GRU for the emotions processing, due to there are just a few (Ekman, 1992) emotions {*disgust, joy, surprise, anger, fear, sadness*} encoded in one hot.
- **Speaker-Specific Emotion Sequence:** This time, they also process the emotions, but concerning the speakers, each speaker has their own GRU for the speaker's emotions
- **Global Speaker Sequence:** For the speakers processing, they also use a Transformer approach encoded in one hot.

The original task in (Kumar et al., 2023a) was to predict the instigator(s) label(s) for each emotion flip; for example, they assign 'nervousness' and 'adoration' instigators to the trigger utterances u_2 and u_3, the instigator is the reason why the emotion flip occurs. We worked on a simple task: identify the trigger in the conversation and which utterance was the cause of the emotion flip.

We propose a new serial of data input to contribute to the model performance; this data is (Mohammad, 2018) NRC VAD Lexicon. This Lexicon contains the {*valence, arousal, dominance*}, with the valence the positive/negative or pleasure/displeasure dimension, arousal is the excited/calm or active/passive dimension. Dominance is the powerful/weak or 'have full control'/' have no control' dimension. We compare the words between the NRC VAD Lexicon dictionary and the words in each utterance and build a personal-

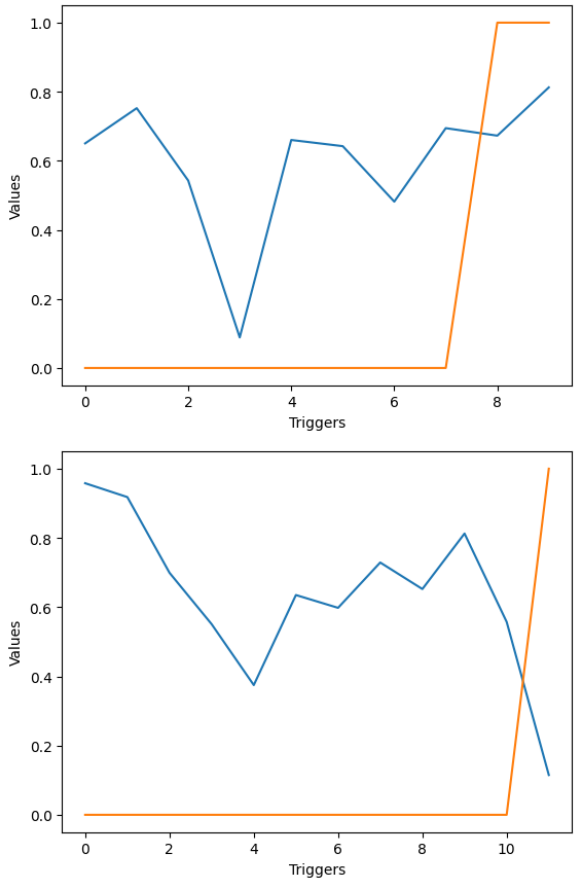


Figure 2: These two graphics describe the relationship between the valence behavior and triggers

ized dictionary for our dataset. We get the following distributions, $u_i = w_0, w_1, ..., w_l$, l begin the number of words in a given utterance, and each word has this distribution $w_l = v_l, a_l, d_l$ (or in the case of the articles, pronouns, etc., the value will be 0), the aim here is to calculate the values of valence, arousal, and dominance that represent the whole utterance, then we apply $1/l \sum_{i=1}^{l} v_i$, $1/l \sum_{i=1}^{l} a_i$, $1/l \sum_{i=1}^{l} d_i$, so the final shape is $u_i = v_{avg}, a_{avg}, d_{avg}$, the VAD values will be lower so we scale it into [0,1]

We took all these features to help the model in the classification task; we found a relationship between the VAD values peak and the trigger sequence location in several samples. see Figure 2

To contribute to the model, we added a new Transformers Encoder for the valence, arousal, and dominance values and a GRU for VAD speaker-specific values. Like the Speaker-Specific Emotion Sequence, we compute a particular speaker's VAD sequence and go through lineal classification layers, see Figure 1.

3.2 Emotion Recognition in Conversation

The architecture for emotion recognition parallels EFR's, with modifications to accommodate emotion as the primary label. This adjustment allows for a direct correlation between VAD scores and emotion classification, eliminating the need for separate emotion modules. The methodological choice to employ GRU models for processing speaker-specific emotional sequences facilitates a refined analysis, enabling the identification of diverse emotional expressions within the conversational context.

4 Experiments

This section presents the experimental setup, including dataset description, data preprocessing techniques, and model evaluation. Utilizing the MELD dataset, we describe our approach to representing conversational utterances through advanced embedding methods. The experiments aim to validate the effectiveness of our methodology in identifying and classifying emotional causes within conversations. Our findings, evaluated against established metrics, indicate a promising direction for future research in multimodal emotion analysis.

4.1 Dataset

MELD. (Poria et al., 2019) is an extension and enhancement of (Chen et al., 2018) EmotionLines. MELD contains dialogues from the TV series Friends. Each utterance is annotated with emotion and sentiment labels and encompasses audio, visual, and textual modalities. The SemEval 2024 Task 10 Subtask 1 presents a variation of MELD, providing *speakers, utterance, emotion* as only textual features and *triggers* as labels.

(Kumar et al., 2023a) identify a set of trigger utterances that cause the emotion to flip at the target. They mark each utterance that acts as a trigger as 'Yes' and the ones not contributing as 'No'.

MaSaC. (Bedi et al., 2023) develop a Hindi-English code-mixed dataset for the multi-modal sarcasm detection and humor classification in conversational dialog. Like MELD, SemEval modifies the dataset for two tasks (ERC and EFR) and only textual data(Kumar et al., 2023c). ERC uses *emotions* as labels, and EFR uses *triggers*

NRC-VAD Lexicon. The National Research Council Canada Valence, Arousal, and Dominance (NRC-VAD) Lexicon (Mohammad, 2018) includes a list of more than 20,000 English words and their valence, arousal, and dominance scores. For a given word and a dimension (V/A/D), the scores range from 0 (lowest V/A/D) to 1 (highest V/A/D). The lexicon with its fine-grained real-valued scores was created by manual annotation using Best-Worst Scaling. The lexicon is markedly more significant than any of the existing VAD lexicons.

4.2 Data Preprocessing

Utterances. To represent the sentences in a dense numerical vector, we use Sentence Embedding pre-trained models. Specifically (Song et al., 2020) the model MPNet, due we handle a sequence of sentences and not a sequence of words is necessary to put all the utterance meaning in just one vector that the Transformer Encoder could process. In the case of Hindi-English code-mixed, we use paraphrase-multilingual-mpnet-base-v2 - Multilingual version of MPNet, trained on parallel data for 50+ languages.

Emotions and Speakers. We use One-Hot Encoder in both cases; the speaker's sequence has a tensor shape of max sequence length and max number of unique speakers in a dialogue in the whole dataset. We assign a one-hot vector for each emotion.

One of the other steps is padding the dataset for every sequence by the maximum sequence length.

5 Results

In our experimental investigations, we meticulously evaluated various configurations of our model and adjusted hyperparameters, alternating between Transformer and GRU modules. For the Emotion Flip Reasoning (EFR) task, we utilized sigmoid neurons with binary cross-entropy loss for binary classification. For Emotion Recognition in Conversation (ERC), we employed softmax neurons with cross-entropy loss for multi-class classification. The F1 score was chosen as the primary metric for evaluation, reflecting the balanced consideration of precision and recall in our assessments.

The integration of Valence, Arousal, and Dominance (VAD) values, crucial emotional dimensions discussed in Section 3, was meticulously analyzed to optimize the model configuration. Drawing on the methodology proposed by (Kumar et al., 2022a), we processed the VAD values with a Transformer Encoder and amalgamated them with other Transformer modules via a linear layer. Similar to processing emotion-specific data, we treated VAD values in a speaker-specific manner using several GRUs, subsequently integrating them with emotion modules through straightforward concatenation.

As depicted in Table 1, the results underscore the significance of including VAD in the model. With VAD integration, the model achieved F1 scores of 0.34 for EFR and 0.23 for ERC, demonstrating improvements of approximately 13% and 5%, respectively, over the configurations without VAD. These findings were consistent across both MELD and MaSaC datasets, highlighting VAD's contribution to enhancing model performance in identifying emotion flips and recognizing emotions in code-mixed conversations.

Our analysis revealed a notable observation regarding the model's distribution loss function. The model's propensity to predict triggers at the beginning of sequences was identified, with ROC and AUC analyses suggesting an optimal threshold of 0.3. For the ERC task, a tendency to predict the 'neutral' category was observed, possibly due to the low deviation of most VAD values from the mean. However, considering the broadly spaced combinations of valence, arousal, and dominance led to slight but discernible improvements in model performance.

Comparatively, while showing promise, our model's performance indicates room for further refinement, especially when juxtaposed with other participants in the shared task. It underlines the necessity for ongoing enhancements and reevaluating the methodological approach to emotion recognition in complex, code-mixed conversational contexts.

Limitations of our current work, including potential dataset biases and the model's generalizability across varied types of code-mixed text, warrant further investigation. Future research directions could encompass exploring additional linguistic features and incorporating other dimensions of emotional reasoning, aiming to build on the foundational insights gained from this study.

6 Conclusion

This study ventured into the complex domain of emotion recognition and reasoning in code-mixed

Model	F1 Score	
	MELD	MaSaC
With VAD	0.34	0.23
Without VAD	0.32	0.20

Table 1: Model Results

conversations, with a particular focus on the tasks of Emotion Flip Reasoning (EFR) and Emotion Recognition in Conversation (ERC). Our primary contribution has been the integration of Valence, Arousal, and Dominance (VAD) values into computational models, aiming to enrich the models' understanding of emotional shifts within dialogues. Despite the modest improvements observed in our experimental results, our work underscores the nuanced challenge of effectively identifying emotion flips and recognizing emotions in code-mixed texts. The incremental advancements achieved, particularly the slight enhancements in F1 scores with VAD values, highlight the potential of incorporating emotional dimensions into NLP models for a deeper understanding of conversational dynamics.

7 Future Work

The pathway forward from this investigation is twofold. Firstly, there is a pressing need to explore additional linguistic and emotional features that could enhance the accuracy and robustness of emotion recognition models. It includes delving deeper into the complexities of code-mixing phenomena and how they influence emotional expression and perception. Secondly, our findings advocate for developing more sophisticated model architectures capable of handling the intricacies of multimodal data and the multifaceted nature of human emotions. Future research should also consider the implications of dataset biases and the challenge of generalizing models across diverse code-mixed contexts. By addressing these areas, subsequent work can build upon the foundational insights provided by this study, contributing to the advancement of NLP and AI's capability to navigate the rich tapestry of human emotions in digital communications.

Acknowledgments

To the SemEval contest, sponsored by the SIGLEX Special Interest Group on the Lexicon of the Association for Computational Linguistics. To the master's degree scholarship program in engineering at the Universidad Tecnológica de Bolivar (UTB) in Cartagena, Colombia.

We would like to express our gratitude to the team at the VerbaNex AI Lab [2] for their dedication, collaboration, and ongoing support of our research endeavors.

References

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2021. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Transactions on Affective Computing*.

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Transactions on Affective Computing*, 14(2):1363–1375.

Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Ting-Hao, Huang, and Lun-Wei Ku. 2018. Emotionlines: An emotion corpus of multi-party conversations.

Lukas Christ, Shahin Amiriparian, Alice Baird, Alexander Kathan, Niklas Müller, Steffen Klug, Chris Gagne, Panagiotis Tzirakis, Lukas Stappen, Eva-Maria Meßner, et al. 2023. The muse 2023 multimodal sentiment analysis challenge: Mimicked emotions, cross-cultural humour, and personalisation. In *Proceedings of the 4th on Multimodal Sentiment Analysis Challenge and Workshop: Mimicked Emotions, Humour and Personalisation*, pages 1–10.

Juan Cuadrado, Elizabeth Martinez, Juan Carlos Martinez-Santos, and Edwin Puertas. 2023a. team utb-nlp at finances 2023: financial targeted sentiment analysis using a phonestheme semantic approach. -.

Juan Cuadrado, Elizabeth Martinez, Anderson Morillo, Daniel Peña, Kevin Sossa, Juan Martinez-Santos, and Edwin Puertas. 2023b. Utb-nlp at semeval-2023 task 3: Weirdness, lexical features for detecting categorical framings, and persuasion in online news. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1551–1557.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & Emotion*, 6:169–200.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024a. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

[2] https://github.com/VerbaNexAI

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023a. Emotion flip reasoning in multiparty conversations. *arXiv preprint arXiv:2306.13959*.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2024b. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, 5(3):1339–1348.

Shivani Kumar, Atharva Kulkarni, Md Shad Akhtar, and Tanmoy Chakraborty. 2022a. When did you become so smart, oh wise one?! sarcasm explanation in multi-modal multi-party dialogues. *arXiv preprint arXiv:2203.06419*.

Shivani Kumar, Ishani Mondal, Md Shad Akhtar, and Tanmoy Chakraborty. 2023b. Explaining (sarcastic) utterances to enhance affect understanding in multimodal dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12986–12994.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023c. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022b. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022c. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Elizabeth Martinez, Juan Cuadrado, Juan Carlos Martinez-Santos, Daniel Peña, and Edwin Puertas. 2023. Automated depression detection in text data: leveraging lexical features, phonesthemes embedding, and roberta transformer model. -.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, Melbourne, Australia. Association for Computational Linguistics.

Luis Gabriel Moreno-Sandoval and Jorge Andres Alvarado-Valencia. 2020. Assembly of polarity, emotion and user statistics for detection of fake profiles. In -.

Luis Gabriel Moreno-Sandoval, Edwin Puertas, Flor Miriam Plaza-del Arco, Alexandra Pomares-Quimbaya, Jorge Andres Alvarado-Valencia, and L Alfonso. 2019. Celebrity profiling on twitter using sociolinguistic. *CLEF (Working Notes)*.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. Meld: A multimodal multi-party dataset for emotion recognition in conversations.

Edwin Puertas and Juan Carlos Martinez-Santos. 2021. Phonetic detection for hate speech spreaders on twitter. -.

Edwin Puertas, Juan Carlos Martinez-Santos, and Pablo Andrés Pertuz-Duran. 2022. Presidential preferences in colombia through sentiment analysis. In *2022 IEEE ANDESCON*, pages 1–5.

Edwin Puertas, Luis Gabriel Moreno-Sandoval, Flor Miriam Plaza-del Arco, Jorge Andres Alvarado-Valencia, Alexandra Pomares-Quimbaya, and L Alfonso. 2019. Bots and gender profiling on twitter using sociolinguistic features. *CLEF (Working Notes)*, pages 1–8.

Edwin Puertas, Luis Gabriel Moreno-Sandoval, Javier Redondo, Jorge Andres Alvarado-Valencia, and Alexandra Pomares-Quimbaya. 2021. Detection of sociolinguistic features in digital social networks for the detection of communities. *Cognitive Computation*, 13:518–537.

Ayan Sengupta, Tharun Suresh, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. A comprehensive understanding of code-mixed language semantics using hierarchical transformer. *arXiv preprint arXiv:2204.12753*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2022. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, pages 1–12.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2023. Are large language models really good logical reasoners? a comprehensive evaluation from deductive, inductive and abductive views. *arXiv preprint arXiv:2306.09841*.

VerbaNexAI Lab at SemEval-2024 Task 3: Deciphering emotional causality in conversations using multimodal analysis approach

Victor Pacheco and **Juan Cuadrado** and **Elizabeth Martinez**
and **Juan Carlos Martinez-Santos** and **Edwin Puertas**
Instituto Tecnologico de Tepic, Tepic México
Universidad Tecnologica de Bolivar, Cartagena Colombia
epuerta@utb.edu.co

Abstract

This study delineates our participation in the SemEval-2024 Task 3: Multimodal Emotion Cause Analysis in Conversations, focusing on developing and applying an innovative methodology for emotion detection and cause analysis in conversational contexts. Leveraging logistic regression, we analyzed conversational utterances to identify emotions per utterance. Subsequently, we employed a dependency analysis pipeline, utilizing SpaCy to extract significant chunk features, including object, subject, adjectival modifiers, and adverbial clause modifiers. These features were analyzed within a graph-like framework, conceptualizing the dependency relationships as edges connecting emotional causes (tails) to their corresponding emotions (heads). Despite the novelty of our approach, the preliminary results were unexpectedly humbling, with a consistent score of 0.0 across all evaluated metrics. This paper presents our methodology, the challenges encountered, and an analysis of the potential factors contributing to these outcomes, offering insights into the complexities of emotion-cause analysis in multimodal conversational data.

1 Introduction

The automatic detection and analysis of emotions in the text have gained substantial traction within the realm of computational linguistics and affective computing, particularly given their profound implications for understanding human-computer interaction (Moreno-Sandoval et al., 2019; Puertas et al., 2021), mental health, and the dynamics of social communication (Cuadrado et al., 2023a; Puertas et al., 2022). Integrating emotion analysis technologies into interactive systems can revolutionize user experience, providing applications that can adapt to users' emotional states in real-time, thus creating more engaging and personalized interactions (Moreno-Sandoval et al., 2020). SemEval-2024 Task 3 introduces a compelling challenge in this arena: the Multimodal Emotion Cause Analysis in Conversations (Wang et al., 2024). This task necessitates the identification of emotions within conversational utterances and elucidating the underlying causes behind these emotional expressions, a nuanced inquiry that extends beyond the textual modality to encompass a multimodal understanding.

In mental health, the automated text analysis for emotional content has shown promise as a tool for early detection of disorders such as depression and anxiety, using data from social media platforms as a rich source of behavioral indicators (Martinez et al., 2023). Similarly, in the domain of social communication, understanding the interplay of emotions and their causes in conversations can enhance the development of systems designed for conflict resolution, educational purposes, and even in the entertainment industry to create more compelling narratives and interactive experiences (Cuadrado et al., 2023b; Moreno-Sandoval et al., 2020; Marrugo-Tobón et al., 2023; Puertas and Martinez-Santos, 2021; Puertas et al., 2019).

Motivated by this task's complexity and innovative potential, our study endeavors to bridge the gap between emotion detection and cause analysis through a methodological approach. By integrating logistic regression models for emotion identification with advanced dependency analysis techniques for cause extraction, we aimed to uncover the intricate patterns of emotional causality in conversations. Our approach is distinguished by its reliance on linguistic features extracted from dependency graphs, facilitating a deeper understanding of the relational structures that underpin emotional discourse.

However, the journey from conceptualization to realization was fraught with challenges. Initial results, marked by a uniform score of 0.0 across evaluation metrics, starkly underscored the task's difficulty and highlighted our methodology's limita-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1339–1343

tions. This outcome reflects the complexities inherent in the nuanced analysis of emotional causality. It emphasizes the need for continuous innovation in the field. The exploration of multimodal data sources, including text, audio, and visual cues, represents a frontier in affective computing, promising to enrich our understanding of emotions in human interactions.

2 Related Work

The task of emotion-cause analysis in conversations has been extensively explored, with recent advancements highlighting the integration of multimodal data and sophisticated natural language processing techniques. This section synthesizes contributions from several seminal works that are directly relevant to the challenges and objectives of SemEval-2024 Task 3 (Wang et al., 2024, 2023; Xia and Ding, 2019).

The MELD dataset introduced by Poria et al. presents a comprehensive multimodal multi-party dataset for emotion recognition in conversations, underscoring the significance of leveraging verbal and non-verbal cues for accurate emotion detection (Poria et al., 2018). This work emphasizes the complexity of emotion and sarcasm recognition in dynamic conversational contexts. It sets a foundational benchmark for subsequent research.

Kumar et al.'s exploration of emotion and its flip in multi-party conversations through a masked memory network and transformer model offers profound insights into the transitional dynamics of emotional states within dialogue (Kumar et al., 2022). Their methodology provides a robust framework for understanding emotional shifts, contributing valuable perspectives to conversational emotion analysis.

Research on multimodal sarcasm detection and humor classification in code-mixed conversations by (Christ et al., 2023) illustrates the challenges and opportunities in identifying nuanced affective states through a combination of textual and visual information. This study highlights the intricate relationship between language use and non-verbal indicators in conveying sarcasm and humor.

The work by (Gupta et al., 2023) on explaining sarcastic utterances to enhance affect understanding in multimodal dialogues further delves into the importance of context and explicit presentation of utterances for sarcasm detection. Their approach underscores the necessity of comprehensive analysis to grasp the underlying affective dimensions in conversations.

The investigation into the reasoning capabilities of large language models by (Xu et al., 2023) questions their efficacy in logical deduction and emotional understanding within conversational settings. This critique prompts a reevaluation of the applications of such models in affective computing, suggesting a need for more nuanced and context-aware methodologies.

Recent studies introduce novel problems and solutions, such as sarcasm explanation in multimodal multi-party dialogues (Yadav et al., 2021), targeted sentiment analysis in the financial domain using transformer-based models and phonestheme embeddings (Vasanth et al., 2022), and automated depression detection leveraging lexical features and RoBERTa transformer models (Naseem et al., 2020). These contributions highlight the expanding scope of affective computing research, including advancements in hate speech detection (Sisi and Jing, 2023) and the nuanced analysis of economic sentiment (Vasanth et al., 2022), underscoring the versatility and complexity of emotion analysis in diverse contexts.

3 Methodology

We designed our methodology to address the challenge of emotion-cause analysis in Conversations, as outlined in SemEval-2024 Task 3, focusing exclusively on textual analysis before considering other modalities. This decision stemmed from our initial goal to establish a solid foundation in text analysis, which, if successful, would have been extended to a multimodal approach. Unfortunately, our endeavors in the textual dimension did not yield the expected results, leading us to concentrate on addressing these challenges without extending our analysis to include audio or video data. We made accessible all scripts and data related to this study at SemEval 2024 VerbaNex AI Repository.

3.1 Data Acquisition and Preprocessing

Our initial phase involved collecting a dataset of conversational utterances provided by the SemEval-2024 Task 3 organizers. Despite the task's multimodal nature, we primarily focused on the textual component. We aimed to extract and analyze linguistic features indicative of emotional expression and causality using standard NLP techniques such as tokenization, lemmatization, and part-of-speech

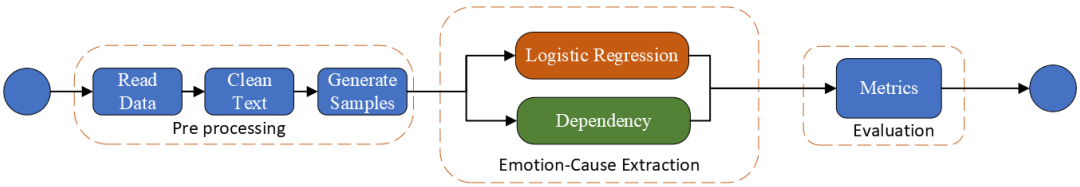


Figure 1: System General Pipeline

tagging facilitated by the SpaCy NLP library.

3.2 Emotion Detection Using Logistic Regression

We implemented a logistic regression model for emotion detection, chosen for its simplicity and effectiveness in binary classification tasks. We trained this model on a dataset subset labeled with emotional annotations. We based the features selected for the model on their potential relevance to emotional expression, including word embeddings derived from pre-trained models and syntactic features extracted during preprocessing. Additionally, we utilized 'CountVectorizer' from scikit-learn to transform text documents into a numerical matrix of word or token counts, which aided in the feature extraction process.

3.3 Dependency Analysis and Feature Extraction

Our core methodology involved analyzing dependency structures within conversational utterances. Utilizing SpaCy's dependency parser and basing our approach on Universal Dependency Relations and the concepts outlined in the "Universal Stanford Dependencies: A cross-linguistic typology," we identified and analyzed noun chunks and their syntactic relationships. We focused on dependencies indicative of emotional causality, such as objects, subjects, adjectival modifiers, and adverbial clause modifiers. These dependency relationships were conceptualized as edges in a graph-like structure, aiding in analyzing the connection between emotional expressions and their causes within the conversation.

3.4 Analysis for Emotion Cause Identification

The culmination of our methodology involved applying a graph-based analysis to map the extracted dependency features onto a framework modeling the conversational flow and interconnections between utterances. This analysis aimed to identify patterns signifying the causality of emotions by tracing the path from an emotional expression to its trigger within the graph. Despite the theoretical soundness of this approach, it encountered challenges in accurately capturing the nuances of emotional causality.

3.5 Task Formulation and Model Inputs

The task was formulated to detect emotion-cause pairs within a complete conversation, with each conversation serving as a single input to our model. We designed this approach to capture the intricacies of conversational dynamics and the contextual interplay of emotional expressions and their causes.

4 Partial Results and Failure Analysis

The logistic regression model achieved a score of 0.52 in emotion detection. However, the final results for emotion-cause pairing were disappointingly at 0.0. The initial low precision of the emotion detector contributed to these results. Further complicating our analysis, handling punctuation and syntactic ambiguities in dependency analysis introduced additional challenges, underscoring the complexity of accurately analyzing emotional causality in text.

5 Conclusions

Using logistic regression and dependency analysis techniques, our exploration into Emotion Cause Analysis in Conversations encountered unexpected hurdles, culminating in a consistent score of 0.0 across all evaluation metrics. This outcome reflects the inherent challenges in interpreting emotional cues from text alone. It highlights a fundamental mismatch between our initial methodological choices and the task's complexity. Our experience underlines the intricacies of detecting and analyzing emotional expressions and their causes within conversational dynamics. Our chosen tools did not capture the nuanced interplay of emotions in a multimodal context.

6 Future Directions

Our research will pivot towards employing more sophisticated models, focusing on transformer-based approaches. These models are renowned for their ability to grasp deeper contextual nuances and hold promise for significantly improving our analysis. Recognizing the pivotal role of multimodal data in enriching emotion analysis, we aim to extend our methodology to incorporate audio and visual cues alongside textual information. This expansion might offer a more holistic understanding of emotional causality in conversations. By refining our evaluation framework and integrating these diverse data types, we aspire to contribute meaningfully to the fields of affective computing and computational linguistics, paving the way for a more nuanced and comprehensive exploration of emotional interactions in conversational settings.

Acknowledgments

To the SemEval contest, sponsored by the SIGLEX Special Interest Group on the Lexicon of the Association for Computational Linguistics. To the master's degree scholarship program in engineering at the Universidad Tecnológica de Bolivar (UTB) in Cartagena, Colombia.

We would like to express our gratitude to the team at the VerbaNex AI Lab [1] for their dedication, collaboration, and ongoing support of our research endeavors.

References

Lukas Christ, Shahin Amiriparian, Alice Baird, Alexander Kathan, Niklas Müller, Steffen Klug, Chris Gagne, Panagiotis Tzirakis, Lukas Stappen, Eva-Maria Meßner, et al. 2023. The muse 2023 multimodal sentiment analysis challenge: Mimicked emotions, cross-cultural humour, and personalisation. In *Proceedings of the 4th on Multimodal Sentiment Analysis Challenge and Workshop: Mimicked Emotions, Humour and Personalisation*, pages 1–10.

J. Cuadrado, E. Martinez, J.C. Martinez-Santos, and E. Puertas. 2023a. Team utb-nlp at finances 2023: Financial targeted sentiment analysis using a phonestheme semantic approach. *CEUR Workshop Proceedings*, 3496.

Juan Cuadrado, Elizabeth Martinez, Anderson Morillo, Daniel Peña, Kevin Sossa, Juan Martinez-Santos, and Edwin Puertas. 2023b. Utb-nlp at semeval-2023 task 3: Weirdness, lexical features for detecting categorical framings, and persuasion in online news. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1551–1557.

Rishabh Gupta, Shaily Desai, Manvi Goel, Anil Bandhakavi, Tanmoy Chakraborty, and Md Shad Akhtar. 2023. Counterspeeches up my sleeve! intent distribution learning and persistent fusion for intent-conditioned counterspeech generation. *arXiv preprint arXiv:2305.13776*.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-Based Systems*, 240:108112.

Duván Andres Marrugo-Tobón, Juan Carlos Martinez-Santos, and Edwin Puertas. 2023. Natural language content evaluation system for multiclass detection of hate speech in tweets using transformers. *In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023).*

E. Martinez, J. Cuadrado, D. Peña, J.C. Martinez-Santos, and E. Puertas. 2023. Automated depression detection in text data: Leveraging lexical features, phonesthemes embedding, and roberta transformer model. *CEUR Workshop Proceedings*, 3496.

Luis Gabriel Moreno-Sandoval, Edwin Puertas, Flor Miriam Plaza-del Arco, Alexandra Pomares-Quimbaya, Jorge Andres Alvarado-Valencia, and L Alfonso. 2019. Celebrity profiling on twitter using sociolinguistic. *CLEF (Working Notes)*.

Luis Gabriel Moreno-Sandoval, Edwin Puertas, Alexandra Pomares-Quimbaya, , and Jorge Andres Alvarado-Valencia. 2020. Assembly of Polarity, Emotion and User Statistics for Detection of Fake Profiles—Notebook for PAN at CLEF 2020. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Usman Naseem, Imran Razzak, Katarzyna Musial, and Muhammad Imran. 2020. Transformer based deep intelligent contextual embedding for twitter sentiment analysis. *Future Generation Computer Systems*, 113:58–69.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Edwin Puertas and Juan Carlos Martinez-Santos. 2021. Phonetic detection for hate speech spreaders on twitter. In *CLEF*.

Edwin Puertas, Juan Carlos Martinez-Santos, and Pablo Andrés Pertuz-Duran. 2022. Presidential preferences in colombia through sentiment analysis. In *2022 IEEE ANDESCON*, pages 1–5.

[1] https://github.com/VerbaNexAI

Edwin Puertas, Luis Gabriel Moreno-Sandoval, Flor Miriam Plaza-del Arco, Jorge Andres Alvarado-Valencia, Alexandra Pomares-Quimbaya, and L Alfonso. 2019. Bots and gender profiling on twitter using sociolinguistic features. *CLEF (Working Notes)*, pages 1–8.

Edwin Puertas, Luis Gabriel Moreno-Sandoval, Javier Redondo, Jorge Andres Alvarado-Valencia, and Alexandra Pomares-Quimbaya. 2021. Detection of sociolinguistic features in digital social networks for the detection of communities. *Cognitive Computation*, 13:518–537.

Wu Sisi and Ma Jing. 2023. Multi-task & multi-modal sentiment analysis model based on aware fusion. *Data Analysis and Knowledge Discovery*, 7(10):74–84.

K Vasanth, P Sridevipon malar, Vikram Shete, CN Ravi, et al. 2022. Dynamic fusion of text, video and audio models for sentiment analysis. *Procedia Computer Science*, 215:211–219.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2023. Are large language models really good logical reasoners? a comprehensive evaluation from deductive, inductive and abductive views. *arXiv preprint arXiv:2306.09841*.

Nikhil Yadav, Omkar Kudale, Aditi Rao, Srishti Gupta, and Ajitkumar Shitole. 2021. Twitter sentiment analysis using supervised machine learning. In *Intelligent data communication technologies and internet of things: Proceedings of ICICI 2020*, pages 631–642. Springer.

VerbaNexAI Lab at SemEval-2024 Task 1: A Multilayer Artificial Intelligence Model for Semantic Relationship Detection

Anderson Morillo and **Daniel Peña**
Juan Carlos Martinez-Santos and **Edwin Puertas**
Universidad Tecnologica de Bolivar, Cartagena Colombia
epuerta@utb.edu.co

Abstract

This paper presents an artificial intelligence model designed to detect semantic relationships in natural language, addressing the challenges of SemEval 2024 Task 1. Our goal is to advance machine understanding of the subtleties of human language through semantic analysis. Using a novel combination of convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and an attention mechanism, our model is trained on the STR-2022 dataset. This approach enhances its ability to detect semantic nuances in different texts. The model achieved an 81.92% effectiveness rate and ranked 24th in SemEval 2024 Task 1. These results demonstrate its robustness and adaptability in detecting semantic relationships and validate its performance in diverse linguistic contexts. Our work contributes to natural language processing by providing insights into semantic textual relatedness. It sets a benchmark for future research and promises to inspire innovations that could transform digital language processing and interaction.

1 Introduction

The analysis of semantic relationships in natural language is considered an essential pillar for understanding the inherent complexity of textual communication Wolfe et al. (2005). With the increasing application of artificial intelligence (AI) models in natural language processing, the ability to discern semantic similarity between text fragments has become a fundamental challenge due to the complexity of natural language and the diversity of meanings that words and phrases can have in different contexts Zunino (2023). In this context, Semantic Textual Relatedness (STR) is a crucial element in natural language understanding, gaining increasing significance with integrating artificial intelligence (AI) models in language processing.

This article aligns with the objectives set by SemEval 2024 Task 1, a pivotal challenge centered on predicting semantic textual relationships between sentence pairs in the English language. The task's importance lies in its profound impact on advancing contextual language understanding, a cornerstone for AI applications across diverse domains.

Our approach to tackle this challenge involves a four-layer feature extraction process. The first layer focuses on extracting lexical similarity, providing a foundation for understanding semantic connections based on word usage. Subsequently, the second layer delves into capturing knowledge-oriented similarity and incorporating domain-specific insights into the model. The third layer concentrates on Corpus-oriented features, considering the contextual influence of larger text corpora. Finally, in the fourth layer, we employ an Embedding approach. Here, we train a Long Short-Term Memory (LSTM) model, extracting sentence features from phoneme embeddings and a sentence transformer model, thereby capturing nuanced semantic nuances.

Throughout the SemEval 2024 Task 1 competition (Ousidhoum et al., 2024b), our system secured the 24th position out of 36 teams, achieving a competitive score of 0.8192. Notably, our system was designed and optimized for the English language. To foster transparency and collaboration, we have released our code, accessible at https://github.com/VerbaNexAI/SemEval2024.

2 Related Work

The analysis of semantic relations is considered fundamental for understanding the connection of meanings between words, phrases, and sentences in a text. Various relationships, such as synonymy, antonymy, hyperonymy, meronymy, and cohyponymy, can manifest in this context. Two main approaches have addressed this field: rule-based and machine learning-based approaches.

Rule-based approaches use ontologies, struc-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1344–1350

tures that define concepts and their relationships, and semantic networks, which are graphical representations of these relationships. Lexical patterns, which are rules that describe semantic relationships based on the structure of words, have also been used. On the other hand, approaches based on machine learning have gained relevance, utilizing technologies such as convolutional neural networks (CNN), recurrent neural networks (RNN), attention models, and word embeddings.

In the literature, we can find several methods for semantic relation detection. In "Learning short-text semantic similarity with word embeddings and external knowledge sources" Nguyen et al. (2019), authors propose an approach that uses word embeddings and external knowledge to measure semantic similarity between short texts, managing to outperform traditional methods on diverse datasets.

Another significant work is "A multi-layer system for semantic relatedness evaluation" Gomaa (2019), which presents a multi-layer system for semantic relatedness evaluation between sentences, combining various similarity features and achieving promising accuracy on the SICK dataset.

In addition, "A New Methodology for Computing Semantic Relatedness: Modified Latent Semantic Analysis by Fuzzy Formal Concept Analysis" Jain et al. (2020) proposes a hybrid methodology that combines latent semantic analysis and fuzzy formal concept analysis to compute the semantic relatedness between words and sentences, obtaining improved results compared to other baseline measures on a specific corpus.

In the field of language-specific semantic relatedness detection, "Sentence Embedding and Convolutional Neural Network for Semantic Textual Similarity Detection in Arabic Language" Mahmoud and Zrigui (2019a) proposes a deep learning-based approach to detect paraphrases in Arabic, using word2vec and a convolutional neural network to overcome traditional methods and other stylometric feature-based approaches.

Finally, "Attention-based model for predicting question relatedness on Stack Overflow" Pei et al. (2021) introduces a deep learning model called ASIM, which uses the attention mechanism to predict the semantic relationship between questions in programming question and answer websites. This model outperforms previous models in terms of performance and generalization in detecting duplicate questions and predicting the relationship between knowledge units.

Despite these advances, knowledge gaps persist. Most models focus on semantic relationships at the word or phrase level. However, we need more research in sentence- and paragraph-level relationship detection. In addition, we require more robust models to adapt to different domains and text types, ensuring a more complete and accurate understanding of semantic relations in natural language processing.

3 System Overview

This section outlines our proposed model for tackling the task presented in SemEval 2024, Track A, which involves assessing the semantic relationship between pairs of sentences. Initially, the text data undergoes preprocessing, including separating sentence pairs, followed by training a Long Short-Term Memory (LSTM) model on the training dataset. Subsequently, we extract text features based on a four-layer architecture proposed by Gomaa (2019), as illustrated in Figure 1. These layers include word embedding, syntactic relationships, corpus topics, and contextual information.

Additionally, we incorporate novel features to enhance our model's performance. These include:

Senticnet: Utilized to extract the polarity of sentences in the knowledge-oriented layer. Latent Semantic Indexing (LSI) is employed for the corpus-oriented layer to gain insights into the underlying structure of the text corpus.

Phoneme Extraction: A novel approach to capture phonetic information from the sentences.

Furthermore, we integrate an attention mechanism inspired by Vaswani et al. (2017) to effectively capture intricate dependencies within sequences. Leveraging insights from recent advancements, our model incorporates a Part-of-Speech (POS)-aware and layer ensemble transformer, further enhancing its ability to discern semantic relationships.

By drawing from diverse studies on data augmentation, ensemble learning, and transformer-based profiling, our model aims to provide a robust solution for semantic relationship detection. It showcases a comprehensive understanding of attention mechanisms and their integration with state-of-the-art techniques.

3.1 Data Description

We used the dataset STR-2022 proposed by Abdalla et al. (2021) and collected by (Ousidhoum et al., 2024a) for training the system. This dataset

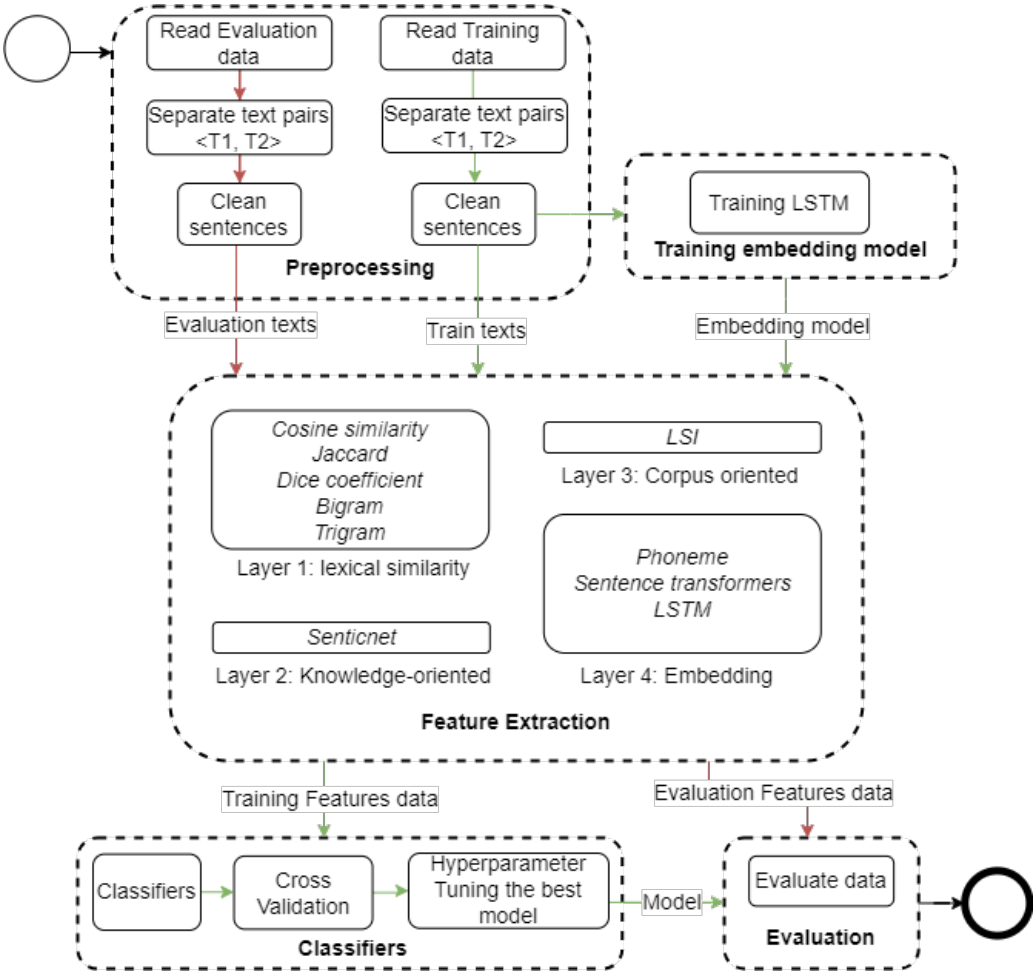


Figure 1: System General Pipeline

comprises 5,500 pairs of sentences in English. This dataset underwent meticulous curation procedures, sampling sentences from various sources, such as social media tweets, book reviews, and paraphrases, to encompass diverse linguistic characteristics and styles. Each pair is labeled with their relatedness score and distribution as shown in Table 1.

Table 1: Frequency distribution of scores intervals

Score intervals	Frequencies
0 - 0.2	502
0.2 - 0.4	1376
0.4 - 0.6	1861
0.6 - 0.8	1149
0.8 - 1	612

3.2 Data Preprocessing

During preprocessing, we separated sentences, and verb and subject decontraction were applied. Subsequently, the model was evaluated with and without lemmatization, as well as with and without stopwords, to assess differences in performance. We removed capitalization, special characters, and numbers as part of the preprocessing process. Data preprocessing is a fundamental step that significantly influences the validity and performance of text classification models, both modern transformers and traditional classifiers Siino et al. (2024). Several preprocessing decisions, such as the treatment of negation, conversion of text to lowercase, application of hyphenation, and consideration of corpus size and document length, are critical to ensure the capture of the true textual meaning and improve the reliability Hickman et al. (2022).

3.3 Training Embedding Models

This part details the training process of a Long Short-Term Memory (LSTM) neural network model for relatedness identification. It begins with data splitting into training and validation sets, followed by message tokenization and constructing a unique vocabulary. We indexed words and applied padding to standardize sequences. We converted

the data into PyTorch tensors and defined a custom dataset and data loaders to handle training batches efficiently.

We defined the model with an embedding layer, an LSTM layer, and a linear output layer. During training, the Adam optimizer and Mean Squared Error (MSE) loss function are utilized, with a loop updating model weights over multiple epochs. For monitoring, we evaluated model performance on the validation set after each epoch.

Finally, upon completion of training, the model and its parameters are saved to a file for future use, enabling its application without the need to retrain it from scratch.

3.4 Feature Extraction

In this section, we explain how the new features are built and used within the text extraction; we try to create a system that could receive the pairs of sentences and return values with consistent output shapes that can feed the model.

3.4.1 Layer 1: String-Oriented Similarity

We based this feature on text extraction, either the characters or the words. It comprises the best features evaluated by Gomaa (2019), Cosine Similarity, Jaccard Similarity, Dice's Coefficient, Bigram, and Trigram.

This layer computes string-oriented similarity features using the following equations:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|} \tag{1}$$

$$\text{Jaccard Similarity} = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

$$\text{Dice's Coefficient} = \frac{2|A \cap B|}{|A| + |B|} \tag{3}$$

where A and B represent the sets of words in both sentences.

3.4.2 Layer 2: Corpus-Oriented Similarity

The system extracts general information using Latent Semantic Indexing (LSI) with the framework Gensim described by Řehůřek and Sojka (2010) to capture the overall thematic similarity between two sentences.

We extract the representation of the average LSI values for the primary five topics, aiming to capture the thematic similarity between the sentences.

3.4.3 Layer 3: Knowledge-Oriented Similarity

This layer is oriented to extract semantic information related to sentiment within the sentences. We used SenticNet proposed by Cambria et al. (2022), a commonsense-based Neurosymbolic framework that extracts the polarity of words. LSI was extracted by averaging the polarities of each sentence and creating a vector with it.

3.4.4 Layer 4: Sentence Embedding

We proposed three forms of word embedding, taking advantage of the good behavior of this layer to evaluate the semantic relationships within two sentences. We used the pre-trained model of sentence transformer Ni et al. (2021), LSTM, and the phonemes embedding. The phoneme embedding works by taking each letter within the sentences, extracting its representation to a phoneme, and returning its representation as a vector.

3.5 Classifiers

We compared the performance of various machine learning models proposed in (Gomaa, 2019) and evaluated the combination of different characteristic types represented by vector inputs.

Random Forest, Gradient Boosting, Multi-layer Perceptron, AdaBoost, and Support Vector Regression (SVR) were employed using the framework Sklearn proposed by Pedregosa et al. (2011), alongside an ensemble voting system combining them. This research aimed to identify the most suitable model(s) for analyzing diverse feature vector inputs.

3.6 Evaluation

The evaluating part of the code serves as a fundamental component within an empirical study focused on assessing the performance of machine learning classifiers. Its primary purpose is to automate the evaluation process, enabling the systematic comparison of various classifiers in a supervised learning context. By implementing k-fold cross-validation, the code ensures robustness and reliability in the evaluation by mitigating potential biases associated with a single train-test split.

Within the evaluating part, we compute a comprehensive suite of performance metrics for each classifier, including Spearman correlation, Mean Squared Error (MSE), R-squared (R^2), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error

(MAPE). These metrics provide a multifaceted assessment of the classifiers' predictive capabilities, accuracy, and robustness.

4 Experimental Setup

The system model utilized sentences without lemmatization and stopword removal, preserving the original form of the sentences to capture a broader range of semantic nuances. Phoneme extraction, as proposed by Del Castillo, was incorporated into the system, utilizing specific components of the provided code. This inclusion aims to enhance the model's sensitivity to phonetic features, contributing to a more comprehensive understanding of textual relationships.

We employ the ShuffleSplit method for cross-validation to assess the model's training. The dataset was split into training and validation sets using a test size 25%, and we utilized ten pairs (n_splits = 10) to ensure robust evaluation. We set the random state to 42 (random_state = 42) for reproducibility. The details of the library versions used in the implementation are provided in Table 2.

Table 2: Python Libraries and Versions

Library	Version
nltk	3.8.1
gensim	4.3.2
spacy	3.7.2
scikit-learn	1.3.2
sentence-transformers	2.2.2
senticnet	1.6
numpy	1.23.4
scipy	1.10.1
matplotlib	3.7.4
seaborn	0.13.0
torch	1.6.0
pandas	2.0.3
epitran	1.24

5 Results

We evaluated the semantic relation detection model using the training and test datasets, and the results are in Table 3. Our model secured the 24th position in the competition ranking, achieving a Spearman correlation coefficient of 0.8192 on the English language dataset. It is relevant to note that this is slightly below the baseline value of 0.83 set as baseline.

Table 3: Ranking of results in framing detection classification

Lang	Spearman Correlation
EN	0.8192

In addition, Table 4 summarizes the performance of the voting system's configuration, and we also present its performance compared to the best individual model.

While the results obtained are below the established baseline, we recognize opportunities for improvement. Abstraction analysis could reveal the specific contributions of each system component to this performance and guide future improvements. It is also crucial to consider the nature of the baseline and the inherent complexity of the task at hand.

6 Limitations

While presenting our approach for evaluating semantic relations between sentences, it's crucial to acknowledge certain limitations that may impact the interpretation and applicability of our proposed model. We outline these limitations below:

- Dataset Representativeness: The STR-2022 dataset, comprising 5,500 English sentence pairs, may not fully capture linguistic diversity and semantic nuances across different languages, limiting the model's generalization to diverse linguistic contexts.
- Preprocessing Impact: Decisions done during preprocessing (such as removing capital letters, special characters, and numbers) could significantly affect semantic representations. When modifying these preprocessing steps, careful consideration is needed to avoid potential bias or information loss.
- Hyperparameter Sensitivity: The model's performance is sensitive to hyperparameter choices, like the number of LSTM layers or the learning rate of the Adam optimizer. Fine-tuning is crucial for optimizing the model's ability to capture semantic relationships effectively.

7 Ethical Considerations

We linked the text similarity field to the detection of paraphrasing (Mahmoud and Zrigui, 2019b), which

Table 4: Correlation of Spearman's Rank between Various Text Preprocessing Methods and Machine Learning Models

Preprocessing	Machine Learning Model	Spearman's Correlation Coefficient
No lemmatized, no stopwords	AdaBoost	0.82
Lemmatized, no stopwords	AdaBoost	0.82
No lemmatized, no stopwords	Gradient Boosting	0.82
Lemmatized, no stopwords	Gradient Boosting	0.82
No lemmatized, no stopwords	Multi-layer Perceptron	0.82
Lemmatized, no stopwords	Multi-layer Perceptron	0.82
No lemmatized, no stopwords	Voting	0.81
Lemmatized, no stopwords	Voting	0.81
No lemmatized, stopwords	Multi-layer Perceptron	0.77
Lemmatized, stopwords	AdaBoost	0.76

can pose an ethical problem when using an author's work without proper citation. Our solution addresses the bias by extracting various text features, from word information to context and vector representation. This way, we can avoid some limitations from training the model with insufficient features.

8 Conclusions

This paper presented a comprehensive approach to evaluating semantic relations between sentences, addressing the challenges posed by SemEval 2024 Task 1. Our model employs a sophisticated four-layered feature extraction technique, encompassing lexical similarity, knowledge orientation, corpus orientation, and embedding layers.

Despite achieving a notable 24th place in the competition, we acknowledge certain limitations, including concerns about dataset representativeness, preprocessing decisions, and hyperparameter sensitivity. These insights serve as valuable lessons for future enhancements in our approach.

While the Spearman correlation of 0.8192 places our model slightly below the established baseline of 0.83, this outcome provides an invaluable learning experience. Moving forward, we plan to conduct ablation studies to dissect the impact of individual components, explore alternative models and preprocessing strategies, and conduct a detailed error analysis to address specific shortcomings.

Ultimately, this work contributes to a deeper understanding of semantic relations and provides a competitive model for SemEval 2024. We are committed to advancing semantic understanding and improving AI systems for natural language processing. The journey from this competition is a stepping stone toward more refined and practical solutions in semantic relationship detection.

Acknowledgments

To the SemEval contest, sponsored by the SIGLEX Special Interest Group on the Lexicon of the Association for Computational Linguistics. To the master's degree scholarship program in engineering at the Universidad Tecnológica de Bolivar (UTB) in Cartagena, Colombia.

We would like to express our gratitude to the team at the VerbaNex AI Lab [1] for their dedication, collaboration, and ongoing support of our research endeavors.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *CoRR*, abs/2110.04845.

Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. 2022. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3829–3839, Marseille, France. European Language Resources Association.

Edwin Alexander Puertas Del Castillo. 2023. *Análisis de elementos fonéticos y elementos emocionales para predecir la polaridad en fuentes de microblogging*. Ph.D. thesis, Pontificia Universidad Javeriana, Colombia 9.

WAEL HASSAN Gomaa. 2019. A multi-layer system for semantic relatedness evaluation. *Journal*

[1] https://github.com/VerbaNexAI

of Theoretical and Applied Information Technology, 97(23):3536–3544.

Louis Hickman, Stuti Thapa, Louis Tay, Mengyang Cao, and Padmini Srinivasan. 2022. Text preprocessing for text mining in organizational research: Review and recommendations. *Organizational Research Methods*, 25(1):114–146.

Shivani Jain, KR Seeja, and Rajni Jindal. 2020. A new methodology for computing semantic relatedness: modified latent semantic analysis by fuzzy formal concept analysis. *Procedia Computer Science*, 167:1102–1109.

Adnen Mahmoud and Mounir Zrigui. 2019a. Sentence embedding and convolutional neural network for semantic textual similarity detection in arabic language. *Arabian Journal for Science and Engineering*, 44:9263–9274.

Adnen Mahmoud and Mounir Zrigui. 2019b. Sentence embedding and convolutional neural network for semantic textual similarity detection in arabic language. *Arabian Journal for Science and Engineering*, 44.

Hien T Nguyen, Phuc H Duong, and Erik Cambria. 2019. Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowledge-Based Systems*, 182:104842.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large dual encoders are generalizable retrievers.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, D. Passos, A. andCournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jiayan Pei, Yimin Wu, Zishan Qin, Yao Cong, and Jingtao Guan. 2021. Attention-based model for predicting question relatedness on stack overflow. In *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, pages 97–107. IEEE.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Marco Siino, Ilenia Tinnirello, and Marco La Cascia. 2024. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. *Information Systems*, 121:102342.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Michael BW Wolfe, Joseph P Magliano, and Benjamin Larsen. 2005. Causal and semantic relatedness in discourse understanding and representation. *Discourse Processes*, 39(2-3):165–187.

Gabriela Mariel Zunino. 2023. Comprender lo desconocido: expectativas, relaciones semánticas y causalidad por defecto revisitada. *Lenguaje*, 51(1):156–186.

UMBCLU at SemEval-2024 Task 1: Semantic Textual Relatedness with and without machine translation

Shubhashis Roy Dipta [†] and **Sai Vallurupalli** [†]
{sroydip1,kolli}@umbc.edu
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250 USA

Abstract

The aim of SemEval-2024 Task 1, "Semantic Textual Relatedness for African and Asian Languages" is to develop models for identifying semantic textual relatedness (STR) between two sentences using multiple languages (14 African and Asian languages) and settings (supervised, unsupervised, and cross-lingual). Large language models (LLMs) have shown impressive performance on several natural language understanding tasks such as multilingual machine translation (MMT), semantic similarity (STS), and encoding sentence embeddings. Using a combination of LLMs that perform well on these tasks, we developed two STR models, *TranSem* and *FineSem*, for the supervised and cross-lingual settings. We explore the effectiveness of several training methods and the usefulness of machine translation. We find that direct fine-tuning on the task is comparable to using sentence embeddings and translating to English leads to better performance for some languages. In the supervised setting, our model performance is better than the official baseline for 3 languages with the remaining 4 performing on par. In the cross-lingual setting, our model performance is better than the baseline for 3 languages (leading to 1^{st} place for Africaans and 2^{nd} place for Indonesian), is on par for 2 languages and performs poorly on the remaining 7 languages.

1 Introduction

The objective of the SemEval 2024 Task 1 is to build and evaluate models capable of identifying relatedness between a sentence pair. Sentence pairs from 14 African and Asian languages belonging to 5 language groups are annotated for relatedness and released for model development. The task is divided into 3 tracks targeting different types of model training: supervised (Track A), unsupervised (Track B), and cross-lingual (Track C). Each track targets a different subset of languages. Extensive details about the languages, language groups, and the data collection process are provided in the task description paper (Ousidhoum et al., 2024a). A detailed description of the shared task, tracks, and datasets are provided in the shared task description paper (Ousidhoum et al., 2024b).

Semantic relatedness helps with understanding language meaning (Jarmasz and Szpakowicz, 2012; Miller, 1995; Antony et al., 2022; Osgood, 1949) and is useful in many areas of natural language processing such as word-sense disambiguation (Banerjee and Pedersen, 2003), machine translation (Bracken et al., 2017) and sentence representation (Reimers et al., 2019; Wang et al., 2022) which have numerous applications. Until recently, semantic relatedness has been mostly restricted to finding word relatedness (Feng et al., 2017; Budanitsky and Hirst, 2006), leading to a lack of sentence-relatedness datasets. At the sentence level, relatedness has been limited to similarity, providing a restricted view of STR (Abdalla et al., 2021). The current shared task aims to broaden the scope of sentence relatedness and extend it to several languages with the goal of encouraging model and resource development in these languages (Ousidhoum et al., 2024b).

Recent advancements in multi-lingual translation and the availability of models for obtaining high-quality sentence embeddings allowed us to explore the effectiveness of machine translated data. Using various sentence embedding models to encode data translated into English, we trained a model, *TranSem*, to find the relatedness score between sentence pairs. Although the task requires the sentence pair to be from the same language, our model can handle sentences from two different languages. Our second model, *FineSem* directly fine-tunes a T5 model (T5 is already fine-tuned on the STS benchmark) on the STR task using both untranslated and translated data to explore the

[†]These authors contributed equally to this work

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1351–1357

usefulness of translation. We use both these models to evaluate languages in Track A. For Track C languages, we use a T5 model fine-tuned only on the english STR data. For evaluating the English dataset in the cross-lingual track, we use a T5 model fine-tuned on the Spanish dataset.

Our contributions to the STR task are as follows: We 1) develop unified models for STR to work with all languages. 2) participate in supervised and cross-lingual tracks. 3) explore the usefulness of machine translation. 4) explore data augmentation using machine translation. Our code is publicly available at https://github.com/dipta007/SemEval24-Task8

2 System Overview

After exploring models and datasets available in the languages we understand[1], we realized the dearth of resources available in these languages. To leverage resources available in English, we translated the 13 non-English languages into English. Assuming the translated data accurately reflects the semantic meaning of the source language, the derived relatedness value from our model for a translated sentence pair should reflect the STR between the sentence pair in the source language. This section describes our machine translation process and models, *TranSem* and *FineSem*. Besides using different training strategies, these models can use both translated and untranslated data.

2.1 Translation to English & Data Augmentation

We use Meta's "No Language Left Behind (NLLB) open-source models" that provide direct high-quality translations for 200 languages with many low-resource languages (Costa-jussà et al., 2022). We use four of the translation models[2] ranging from 600 million to 3.3 billion parameters. Using each of the 4 models, we translated the training data for all languages in track A, except Amharic and Algerian Arabic, and obtained 4 translated datasets for each language. None of the 4 models we used supported Amharic, Algerian Arabic or Punjabi. We decided against translating track C languages with 3 of 12 unsupported languages. Using 4 different model translations gave us a 4-fold augmentation of the training data. We translated the test data using only the largest model (*facebook/all-200-3.3 B*) to obtain the best translation features.

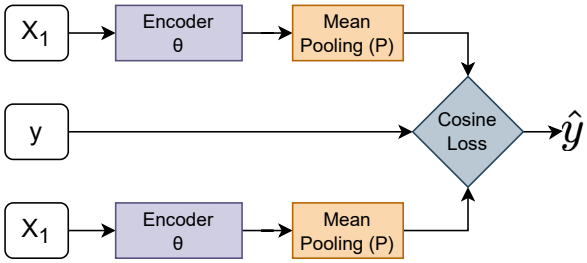


Figure 1: Overview of TranSem model architecture (Inspired by Reimers et al. (2019)). The encoder (θ) is shared, and the diamond box represents the loss function. The encoded sentence pairs (x_1, x_2) and the label (y) are the input to the cosine similarity loss.

2.2 *TranSem* Model

Inspired by Reimers et al. (2019), we used the Siamese model architecture (shown in Fig. 1). For a given pair of sentences (x_1, x_2) and their semantic relatedness score (y), we encode each sentence with a sentence encoder (θ). The embeddings for the sentences go through a pooling operation (P) to produce sentence embeddings (s_1, s_2). The cosine similarity of the encoded embeddings is trained to match the semantic relatedness score using the mean-squared error loss:

$$\mathcal{L} = MSE(\text{cos-sim}(P\theta(x_1), P\theta(x_2)), y) \quad (1)$$

We experimented with several sentence encoding models for encoding our translated and augmented training dataset. We chose DistilRoberta[3] to submit results for the competition leaderboard based on our primary validation (details on §4.1.1, §4.1.2, and §4.1.3). This is a distilled version of RoBERTa (Liu et al., 2019) fine-tuned on sentence-level datasets and suitable for clustering and semantic searches, which we further fine-tuned on our translated and augmented dataset. The sentence-t5-xl embedding model was chosen to compare the effectiveness of sentence embeddings as opposed to the direct fine-tuning used in the *FineSem* model. After experimenting with different pooling mechanisms of mean, max, and CLS tokens, we found that mean pooling works well for our setting. This aligns with earlier findings, which show that mean pooling produced encodings lead to better performance on downstream tasks.

[1] Of the 14 languages, the authors are proficient in English, Hindi & Telugu languages.

[2] facebook/nllb-200-3.3B, facebook/nllb-200-1.3B, facebook/nllb-200-distilled-1.3B, facebook/nllb-200-distilled-600M

[3] sentence-transformers/all-distilroberta-v1

2.3 *FineSem* Model

T5 (Raffel et al., 2020) is a transformer model that uses transfer learning; the model trained on "Colossal Clean Crawled Corpus" is fine-tuned on a mixture of 8 downstream unsupervised and supervised tasks converting them into a unified text-to-text task setting. The T5 model is available in several sizes, of which we use the base, large, and XL models ranging from (660 million to 3 billion parameters). One of the supervised tasks used in the T5 model training is the semantic textual similarity benchmark (STS-B) dataset trained as a regression classification problem. We use the STS task setting to train on the track A STR training datasets using 3 different options: separate T5 models trained on individual languages, a single model trained on all 14 languages (without translation), and a single model trained on the translated and augmented dataset (for 12 languages). These settings allow us to contrast the effectiveness of direct fine-tuning with the sentence embedding-based *transem* model and the usefulness of machine translation.

From the T5 models fine-tuned on the individual languages, we use the English and Spanish fine-tuned models (we refer to these models as the English and Spanish models) for evaluating the cross-lingual Track C languages. We use the English model to evaluate development and test data from all languages except English and the Spanish model to evaluate the English data.

3 Experimental Setup

3.1 HyperParameters

We train our models using AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 1e-5 and weight decay of 0.01. We use an effective batch size of 32 (batch size 16 with 2 steps of gradient accumulation) (for TransSem) and a batch size of 16 (for FineSem).

We train our *transem* model infinitely with an early stopping patience of 10 on the validation Spearman Correlation score. We train the *finesem* model for 10 epochs (2 epochs for the model trained on the translated and augmented model) and checkpoint the models at the end of every epoch. We evaluate the dev sets for each language against these 10 checkpoints. We evaluate the corresponding test data using the checkpoint which provides the best performance on the dev data for a language.

3.2 Infrastructure

All experiments were conducted on an NVIDIA Quadro RTX 8000 with 48GB of VRAM and A100 80GB. We utilize the PyTorch Lightning library [4] for conducting the experiments and Weight & Biases [5] for logging purposes (for the *TransSem* Model) and HuggingFace Transformers (for the *FineSem* Model).

4 Results

We first report our results and analysis on Track A languages (§4.1), and then on Track C (§4.2). We use the official baseline (Ousidhoum et al. (2024b)) that used LaBSE (Feng et al., 2020) fine-tuned on the provided training dataset and refer to this model as $baseline$.

4.1 Track A Languages

This section discusses our findings using various model settings with the *TranSem* model.

4.1.1 Effect of Batch Size

Comparing performance with various batch sizes (results are shown in Table 1), we show that our batch size selections are fairly good (32 for task *TranSem* and 8 for *FineSem*).

	Batch Size	A3 eng	A4 hau	A5 kin	A6 mar	A7 ary	A8 esp	A9 tel	avg
TranSem+	2	.8102	**.6857**	.6886	**.8515**	.7376	**.6519**	**.8342**	**.7514**
TranSem+	4	.5415	.1355	.0643	.5394	.3038	.5998	.4231	.3725
TranSem+	8	.8132	.6519	.6642	.8348	.7340	.6355	.8187	.7360
TranSem+	16	.8093	.6377	**.6997**	.8429	**.7500**	.6462	.8257	.7445
TranSem+	64	.8092	.6589	.6456	.8271	.7247	.6234	.8131	.7289
TranSem+	128	.8129	.6787	.6659	.8417	.7411	.6396	.8324	.7446
TranSem+	256	**.8152**	.6716	.6589	.8357	.7197	.6353	.8189	.7365

Table 1: The effect of batch size on $TranSem$ for different batch sizes $\{2, 4, 8, 16, 64, 128, 256\}$

4.1.2 Effect of Encoder Pooling

In Table 2, we compare performance using 3 different pooling operations. We used mean pooling for the official results we submitted, as it showed good performance in most languages.

	Pooling	A3 eng	A4 hau	A5 kin	A6 mar	A7 ary	A8 esp	A9 tel	avg
TranSem+	CLS	**.8133**	**.6737**	.6655	.8339	.7376	.6363	**.8309**	.7416
TranSem	Mean	.8125	.6403	**.6807**	**.8406**	**.7448**	**.7211**	.8255	**.7522**
TranSem+	Max	.7960	.6157	.5809	.8227	.6643	.6075	.7997	.6981

Table 2: The effect of pooling on $TranSem$ using different pooling mechanisms (CLS Token, Mean, Max)

[4] https://lightning.ai/
[5] https://wandb.ai/

4.1.3 Effect of Sentence Embedding Models

In Table 3, we provide contrastive results with several sentence embedding models used in *TranSem*. For official results, we submitted results from the distilroberta-v1 sentence embedding model (results for some of the languages are from the *FineSem* model fine-tuned on individual STR training datasets).

4.1.4 Usefulness of Machine Translation and Direct Fine-tuning

We compare the performance of the *FineSem* models fine-tuned using the 3 data options (results are shown in Table 3). FineSem-Individual shows the performance of T5-XL models fine-tuned on the individual datasets. Unified and Translated models show the performance of the two T5-XL models fine-tuned on the untranslated data and translated+augmented data. The model trained on untranslated data performs poorly on the Marathi dataset, but performs on par with the other models indicating that we may not need to translate all languages to English. We find that direct fine-tuning with the translated and augmented data is comparable with the *TranSem* model using various sentence embeddings.

4.2 Track C Languages

In Table 4, we compare the performance of various T5 models on the track C languages. We submitted official results using our T5-XL based FineSem model (FineSem-LB) where the results are obtained using the checkpoint after the third epoch. With the same model we also report with the approach where we use the checkpoint which results in the best performance on the development data for a given language. These results are shown as FineSem-XL. We compare these results with the T5-base and T5-large based FineSem models. We bold the best scores for easy readability but underline scores that are better than the baseline. Among our models the overall performance of the XL model is better and this model improves upon the baseline for Afrikaans, Indonesian and Spanish.

5 Related Work

In this section, we present previous research conducted in the fields of Machine Translation (see §5.1), Sentence Embedding (see §5.2) and Semantic Similarity (see §5.3).

5.1 Machine Translation

Machine translation has evolved in the last 75 years from rule-based systems to statistical-based systems to the current neural machine translation (NMT) systems. In the 10 years since the first sequence-to-sequence NMT model (Bahdanau et al., 2014), machine translation reached a point where translations from models for high-resource languages rival human translators (Läubli et al., 2018; Popel et al., 2020). This was possible due to the amount of bilingual data pairs available for training in these languages (Haddow et al., 2022). Translation systems for medium and low resource languages that lacked the scale of these resources either developed cross-lingual models (Nguyen and Chiang, 2017; Zoph et al., 2016) or developed datasets (Bañón et al., 2020; Schwenk et al., 2019). Current state-of-the-art translation models use a many-to-many approach to handle a large number of medium to low-resource languages (Costa-jussà et al., 2022).

5.2 Sentence Embedding

Generating a sentence-level embedding is useful for semantic searches and clustering. Since the first transformer model (Vaswani et al., 2017), several encoder-only models such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2018) were used to learn effective sentence embeddings that also performed well on downstream NLP tasks (Cer et al., 2018; Roy Dipta et al., 2023). Reimers et al. (2019) developed a Siamese-like network architecture with two BERT sentence embedding models that improved semantic search systems. Using T5 (Raffel et al., 2020) models in a similar architecture, sentence embeddings produced by T5 were shown to be superior to the encoder-only model embeddings with performance gains in downstream tasks (Ni et al., 2021). A more recent work (Reimers and Gurevych, 2020) showed that a teacher-student model can be efficiently used to develop a sentence embedding system for many low-resource languages.

5.3 Semantic Similarity

Semantic textual similarity captures a type of semantic relatedness requiring similarity on all aspects between a sentence pair. SemEval tasks on semantic textual similarity from 2012 to 2017 resulted in the STS benchmark (Cer et al., 2017). Recently, Deshpande et al. (2023) proposed condi-

Model	A3 eng	A4 hau	A5 kin	A6 mar	A7 ary	A8 esp	A9 tel	avg
baseline	**.8300**	**.6900**	**.7200**	**.8800**	**.7700**	.7000	.8200	**.7729**
distilroberta-v1 (TranSem)	.8125	.6403	.6807	.8406	.7448	**.7211**	.8255	.7522
mpnet-base-v2	.8104	.6692	.6971	**.8568**	.7297	.6518	.8250	.7486
roberta-large-v1	**.8260**	.6750	**.7056**	.8461	.7480	.6298	**.8394**	**.7528**
sentence-t5-xl	.8236	.6440	.6720	.8324	.7124	.6277	.8250	.7339
multi-qa-mpnet-base-dot-v1	.8111	**.6852**	.7041	.8482	.7163	.6586	.8245	.7497
all-MiniLM-L12-v2	.8237	.6474	.7026	.8522	**.7605**	.6141	.8247	.7465
FineSem-Individual	.8385	.6335	**.7175**	.2211	**.7647**	.6900	.6085	.6391
FineSem-Unified	**.8438**	.6369	.6837	.3878	.6265	**.7040**	.6993	.6546
FineSem-Translated	.8105	**.6383**	.7133	**.8608**	.7403	.6663	**.8152**	**.7493**

Table 3: Model Performance (Spearman Correlation Coefficient) on Subtask A test set. $TranSem$ shows results submitted before the official deadline, $baseline$ shows official baseline results, and the rest are contrastive results for our various models. The best scores within each section are **bolded**, and best scores across all sections are **underlined**.

Model	C1 Afr	C2 Arq	C3 Amh	C4 Eng	C5 Hau	C6 Hin	C7 Ind	C8 Kin	C9 Arb	C10 Ary	C11 Pan	C12 Esp	avg
Baseline	.7900	**.4600**	**.8400**	**.8000**	**.6200**	**.7600**	.4700	**.5700**	**.6100**	**.4000**	**-.050**	.6200	.5742
FineSem-LB	**.8223**	1263	.0430	.7875	.4569	.1552	.5153	.4836	.0354	-.0375	-.0775	.6089	**.3266**
FineSem-XL	.8164	1023	.0373	**.7889**	.4561	.1594	**.5279**	.4128	.0000	.0219	-.0817	.6259	.3246
FineSem-L	.8007	-.0515	.0112	.7752	**.4831**	.1764	.4419	**.5094**	.0154	**.0331**	-.0591	.6605	.3164
FineSem-B	.7802	**.1799**	**.2543**	.7448	.4784	**.2404**	.4517	.3861	**.0527**	.0268	**-.0520**	**.6289**	.3477

Table 4: Model performance (Spearman Correlation Coefficient) on Track C test sets. All language test sets (except English) use the FineSem models trained on the English training set. The English test set uses the FineSem models trained on the Spanish training set. The best scores among our models are **bolded**. Scores better than baseline are **Underlined**.

tional semantic textual similarity to explore semantic relatedness.

6 Conclusion & Future Work

We developed two different models and showed how the models performed in supervised and cross-domain training tasks in 14 languages. We explored using machine translation, sentence encoders, and SST-B style training with T5 models. Our models improved over the official baseline for some of the languages. For computational purposes, we have excluded using more recent models like mistral-7b (Jiang et al., 2023), which have outperformed most of the open-source and close-source models in various benchmarks (Zheng et al., 2024). For future work, we intend to explore prompting for STR and prompt-based LLMs [6] for translation.

7 Disclaimer

We did not use AI assistants to write any part of our paper or code. All writing is original and produced by the authors.

[6] https://chat.openai.com/

8 Limitations

We acknowledge our work has the following limitations. We use several pre-trained LLMs in our experiments. It is well known that these models can echo biases and misinformation either implicitly or explicitly. We did not control for any of these when training them on the STR datasets. In addition, the STR datasets may also echo several biases related to social groups, cultural groups, race, gender, behavioral, and perceptual differences of annotators. We did not explore or control for any of these biases in our work. As a result, our work carries the limitations of both the models and the datasets we used.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *CoRR*, abs/2110.04845.

James W Antony, America Romero, Anthony H Vierra, Rebecca S Luenser, Robert D Hawkins, and Kelly A Bennion. 2022. Semantic relatedness retroactively boosts memory and promotes memory interdependence across episodes. *eLife*, 11:e72519.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, page 805–810, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. 2020. Paracrawl: Web-scale acquisition of parallel corpora. Association for Computational Linguistics (ACL).

Jennifer Bracken, Tamar Degani, and Natasha Tokowicz Chelsea Eddington. 2017. Translation semantic variability: How semantic relatedness affects learning of translation-ambiguous words. *Bilingualism: Language and Cognition*, 20(4):783–794.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Ameet Deshpande, Carlos E. Jimenez, Howard Chen, Vishvak Murahari, Victoria Graf, Tanmay Rajpurohit, Ashwin Kalyan, Danqi Chen, and Karthik Narasimhan. 2023. C-sts: Conditional semantic textual similarity.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Yue Feng, Ebrahim Bagheri, Faezeh Ensan, and Jelena Jovanovic. 2017. The state of the art in semantic relatedness: a framework for comparison. *The Knowledge Engineering Review*, 32:e10.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732.

Mario Jarmasz and Stanialaw Szpakowicz. 2012. Roget's thesaurus and semantic similarity. In *Recent Advances in Natural Language Processing*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41.

Toan Q Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. *arXiv preprint arXiv:1708.09803*.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *CoRR*, abs/2108.08877.

C. E. Osgood. 1949. The similarity paradox in human learning: a resolution. 56:132–143.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima

Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. 11.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers, Nils Reimers, Iryna Gurevych, and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv: Computation and Language*.

Shubhashis Roy Dipta, Mehdi Rezaee, and Francis Ferraro. 2023. Semantically-informed hierarchical event modeling. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 353–369, Toronto, Canada. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Bin Wang, C.-C. Jay Kuo, and Haizhou Li. 2022. Just rank: Rethinking evaluation with word and sentence similarities. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6060–6077, Dublin, Ireland. Association for Computational Linguistics.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. In *Proceedings of the Third Workshop on Representation Learning for NLP*, pages 164–174, Melbourne, Australia. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.

MasonTigers at SemEval-2024 Task 9: Solving Puzzles with an Ensemble of Chain-of-Thought Prompts

Md Nishat Raihan, Dhiman Goswami, Al Nahian Bin Emran,
Sadiya Sayara Chowdhury Puspo, Amrita Ganguly, Marcos Zampieri
George Mason University, USA
mraihan2@gmu.edu

Abstract

This paper presents the *MasonTigers*' submission to the SemEval-2024 Task 9 which provides a dataset of puzzles for testing natural language understanding. We employ large language models (LLMs) to solve this task through several prompting techniques. We show that zero-shot and few-shot prompting with proprietary LLMs outperform open-source models. Results are further improved with chain-of-thought prompting. We obtain our best results by utilizing an ensemble of chain-of thought prompts, ranking 2^{nd} in the word puzzle subtask and 13^{th} in the sentence puzzle sub-task.

1 Introduction

In recent years, LLMs have achieved impressive performance on several question answering and language understanding tasks when provided with appropriate prompting (Brown et al., 2020). However, complex reasoning abilities often present a challenge for these models. SemEval-2024 Task 9 (Jiang et al., 2024b) introduces a novel dataset called *BrainTeaser* (Jiang et al., 2023) which includes a set of complex puzzles and brainteasers. Such tasks involve solving word and sentence puzzles, which require multi-step inference and deduction. The dataset covers a diverse range of puzzle types including sequences, analogies, classification, mathematical reasoning, inferences about implicit relationships, and more. Solutions frequently demand a chained application of knowledge and logic across multiple steps to uncover insights or concepts not directly stated in the problem description.

Solving these elaborate reasoning problems is a challenging scenario for NLP systems. We explore whether and how LLMs can succeed on this task. We employ proprietary models such as GPT-4 (OpenAI, 2023) and Claude 2.1 (Anthropic, 2023) through APIs. These models have shown promising few-shot reasoning ability. We also use Mixtral (Jiang et al., 2024a), an open-source LLM that shows state-of-the-results in several language reasoning tasks. The prompting paradigm involves providing models with natural language descriptions that encode the reasoning process step-by-step (Liu et al., 2021). We implement various prompting approaches for mapping puzzles to conditional text and systematically transforming reasoning into explanation chains. Our core method, chain-of-thought prompting (Wei et al., 2022), iteratively breaks down the deduction into simplified logical steps.

Experiments reveal that while zero-shot performance lags due to a lack of grounding, multi-step prompts can unlock substantial reasoning ability in models. Performance improves with more steps and more specificity in the prompt. While introducing few-shot prompting generates good results, we observed that models do significantly better with chain-of-thought prompting. We experiment with several chains of thought and achieve mostly similar results with each attempt. To make a more empirically confident guess towards solving the puzzles we adopt an ensemble of these chains based on majority voting. Our approach achieves competitive performance, ranking *2nd* on the word puzzle subtask and *13th* on sentence puzzles.

2 Related Work

LLMs have been widely used for complex and challenging language processing tasks recently (Raihan et al., 2023a,b; Goswami et al., 2023). They have shown good reasoning abilities in several tasks. The task of solving puzzles and the BrainTeaser dataset (Jiang et al., 2023) represent both a novel task and a novel dataset respectively. Similarly to their multiple choice questions (MCQs) approach, a few datasets like MathQA (Austin et al., 2021), have been compiled. However, these are intended for specific tasks in which domain knowledge is usually enough thus they not requiring deep reason-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1358–1363

ing. A similar work is done by Saeedi et al. (2020) where they investigate a task that combines natural language understanding and commonsense reasoning. They present deep learning architectures for distinguishing between sensible and nonsensical statements.

Pun detection by (Zou and Lu, 2019) is a puzzle-like activity that is similar to BrainTeaser. It presents a method for joint pun detection and localization utilizing a sequence labeling perspective. This highlights the complexity of language comprehension, especially in detecting subtle wordplay. Another dataset, LatEval is curated by Huang et al. (2023) that delves further into lateral thinking and commonsense reasoning, highlighting the challenges faced by language models in tasks requiring unconventional thinking and creativity. Zhou et al. (2023) presents ROME, a dataset designed to assess vision-language models' capacity to reason beyond intuitive understanding, highlighting the shortcomings of existing models in understanding events that defy common sense.

In the field of reasoning task, a *chain-of-thought* (Wei et al., 2022) implies a logical sequence of connected ideas, fostering coherence and depth in responses. On the other hand, a *tree-of-thought* suggests branching out into various related ideas, offering a more comprehensive exploration of a topic. While few-shot prompting is effective for some tasks by providing examples to guide the model, it may have limitations in capturing the complexity of nuanced conversations. The optimal choice may involve a hybrid approach, where a few-shot prompt sets the initial (Yao et al., 2023) context, and the model subsequently follows a chain or tree of thought to generate more contextually rich and coherent responses useful for reasoning tasks.

Tan (2023) shows the performance of LLM's on the reasoning of arithmetic word problems. It states that higher degrees of realization are associated with better overall accuracy on arithmetic problems. And chain-of-thought is really helpful in this aspect as it covers a variety of prompts to strengthen the reasoning. Similarly, Mo and Xin (2023) presents a new reasoning framework for large language models by addressing a gap in prior tree-based reasoning methods which overlooked inherent uncertainties in intermediate decision points made by models. Overall, the key innovation is leveraging uncertainty estimation locally within the models during tree reasoning to enable more precise problem-solving and reasoning.

3 The BrainTeaser Dataset

The BrainTeaser dataset (Jiang et al., 2023), introduced with the task (Jiang et al., 2024b) is a question-answering benchmark designed to evaluate models' ability for lateral thinking, i.e., to defy default commonsense associations and reason creatively. The dataset contains 1,100 multiple-choice questions divided into two sub-tasks - 627 sentence-based puzzles relying on narrative context and common phrases and 492 puzzles focused on the literal form and letters of words.

For a fair comparison with human performance, the dataset also provides a separate human evaluation set with 102 randomly sampled questions. Each question in BrainTeaser has one correct answer and three distractor choices, including the option "none of the above". To prevent memorization of training data, the dataset also contains semantically and contextually reconstructed variants for every question while preserving the original reasoning process and answers. The key statistics of the dataset are shown in Table 1.

	Sentence	**Word**
Number of puzzles	627	492
Avg. tokens (prompt)	34.88	10.65
Avg. tokens (choices)	9.11	3.0

Table 1: Key statistics of the BrainTeaser dataset in the sentence and word puzzle sub-task.

During the SemEval-2024 Task 9 development phase, a total of 240 prompts (120 for both sentence and word puzzles) are provided. During the test phase, a total of 216 prompts (120 for sentence and 96 for word puzzles) are provided.

4 Experiments

In our experiments, we focus on several prompting strategies by employing three state-of-the-art models including proprietary models like GPT-4 (OpenAI, 2023) and Claude 2.1 (Anthropic, 2023) (accessed via API key) and one open-source model - Mixtral (Jiang et al., 2024a).

4.1 Zero-Shot Prompting

We start with zero-shot prompting by assigning the AI a role, describing the task, and giving it one puzzle at a time, as shown in Figure 1.

Role:
You are a helpful AI assistant. You are given the task of solving a sentence/word puzzle.

+

Definition:
It's a Multiple Choice Question paired with 4 potential answers. Choose the most suitable one.

+

Question: What part of London is in France?
Choices:
1. The letter N.
2. The letter O.
3. The letter L.
4. None of the above.

Choose the most suitable answer. Thanks.

Figure 1: Sample structure for Zero-Shot Prompting.

4.2 Few-Shot Prompting

In order to give the LLMs more context we integrate more examples and design prompts for few-shot prompting. We include 4 solved puzzles from the train set and then attach one puzzle from the test set each time we prompt the models. We also use some tags for better extracting the generated answers, as shown in Figure 2.

4.3 Chain-of-Thought

To guide the models toward better reasoning - we experiment with chain-of-thought prompting. We give the model the puzzle, and the potential answers and work with every example one-by-one in order to choose the most reasonable one. Like the original CoT approach (Wei et al., 2022), we do not assign any role or explain the task - just pose the question, the CoT, and the answer (see Figure 3). We do this as 2-shot, 4-shot, and 8-shot for all three models.

4.4 Ensemble of Chain-of-Thought Prompts

To assess model performance, an ensemble approach is utilized with chain-of-thought prompting to make more confident guesses regarding the correct answers. Specifically, majority voting is done across an ensemble of models prompted by different question groups. For each prompt, 8 different random questions are selected from the BrainTeaser training set - repeated 5 times in total. Finally, the predictions are aggregated through voting to output the overall ensemble prediction.

This ensemble methodology with chain-of-thought prompting helps improve robustness to outlier examples and noise compared to using a single model. By prompting the ensemble components on different random question subsets, diversity is promoted to capture a more holistic representation of the overall data distribution. The voting also helps cancel out issues with single models latching onto spurious patterns. Overall, the ensemble approach with multiple chain-of-thought prompt sets provides a robust assessment strategy suited for the open-ended nature and diversity of the BrainTeaser puzzles.

Role:
You are a helpful AI assistant. You are given the task of solving a sentence/word puzzle.

+

Definition:
It's a Multiple Choice Question paired with 4 potential answers. Choose the most suitable one.

+

Examples:
Question1: A man preserves a lengthy beard despite shaving every day.
1. He is a barber.
2. He wants to maintain his appearance.
3. He wants his girlfriend to buy him a razor.
4. None of the above.
Correct Answer: <ans1> 1 **</ans1>**
.....
Question4: Tom attends class every day but doesn't do any homework.
1. He is a teacher.
2. He is a lazy person.
3. His teacher will not let him fail.
4. None of the above.
Correct Answer: <ans4> 1 **</ans4>**

+

Question5: The brother of a beggar passed away, but the deceased had no brothers. How is that possible?
Choices:
1. The beggar was the man's sister.
2. The man is angry for his brother being a beggar and cut ties with him.
3. The bagger's brother is a murderer.
4. None of the above.
Correct Answer: <ans5> ? **</ans5>**
Choose the most suitable answer. Thanks.

Figure 2: Sample structure for Few-Shot Prompting.

5 Results

We analyze the performance of the three models - including GPT4 Turbo, Claude 2.1, and Mixtral.

Question 1: How do you spell COW in thirteen letters?

–

Choices 1:

1. SEE OH DEREFORD.
2. SEE O DOUBLE YOU.
3. COWCOWCOWCOWW.
4. None of above.

–

Chain-of-Thought 1:

1. SEE OH DEREFORD: Doesn't seem to spell out "COW" in any conventional or playful manner.
2. SEE O DOUBLE YOU: Spells out "COW" in a creative way, matching the letter count required.
3. COWCOWCOWCOWW: Simply repeats the word "COW" without cleverly meeting the thirteen-letter criteria.
4. None of the above: Not applicable since there is a viable option.

–

Decision 1: The answer "SEE O DOUBLE YOU" creatively spells "COW" using thirteen letters, making it the correct choice.

–

Answer 1: 2.

.....

.....

.....

.....

Question 8: –

Choices 8: –

Chain-of-Thought 8: –

Decision 8: –

Answer 8: –

———

———

Question 9: How do you spell COB in seven letters?

–

Choices 9:

1. COBCOBB
2. COBBLER
3. SEE O BEE.
4. None of the above.

–

Figure 3: Sample structure for Chain-of-Thought Prompting (8-shot).

These models are tested with different types of prompts - regular and chain-of-thought, and with a varying number of examples, known as shots, ranging from zero to eight. Additionally, an ensemble method is applied to the eight-shot chain-of-thought prompting to see if it can further improve the models' performance. The results, shown in Table 2, reveal how the models performed under each condition. A human baseline with scores of 0.91 for both Sentence and Word puzzles in the test set is provided by the task organizers for comparison purposes.

GPT4 Turbo shows the best performance, especially with chain-of-thought prompting and an increasing number of shots. The model performs best with the eight-shot chain-of-thought prompting combined with the ensemble method ([E]), reaching the highest Sentence and Word scores of 0.93 and 0.95 in the test set, respectively. This shows that chain-of-thought prompting and the ensemble method significantly improve the model's understanding and output. Claude 2.1 also improves with chain-of-thought prompting and more shots. Its best scores were with the eight-shot chain-of-thought with the ensemble, achieving Sentence and Word scores of 0.86 and 0.95 in the test set, respectively. The asterisk (*) mark in Table 2 denotes our submission during the test phase. Even though Mixtral's performance is inferior to the performance of the other two models, it consistently gets better with more shots and chain-of-thought prompting. Mixtral delivered best results with the eight-shot chain-of-thought and the ensemble technique, with Sentence and Word scores of 0.88 and 0.82 in the test set, respectively.

Finally, the results highlight the effectiveness of chain-of-thought prompting in boosting the performance of LLMs. This approach, especially when combined with more examples and the ensemble method, greatly improves models' abilities to process and generate more accurate responses. GPT4 Turbo's top performance is likely due to its advanced design, which makes the most of these strategies. On the other hand, Claude 2.1's results point to the importance of model-specific adjustments.

6 Conclusion and Future Work

In this paper, we presented MasonTigers' approach to SemEval-2024 Task 9 on solving puzzles using LLMs. We explored various prompting strategies to guide the models, including zero-shot, few-shot, and chain-of-thought prompting. Our key method involved iteratively breaking down reasoning into simplified logical steps to decompose the complex

Model	Prompting	# of Shot	Sen_Dev	Sen_Test	Word_Dev	Word_Test
Human Baseline	–	–	–	0.91	–	0.91
GPT4 Turbo	Regular	Zero Shot	0.79	0.76	0.81	0.79
GPT4 Turbo	Regular	4 Shot	0.90	0.91	0.87	0.86
GPT4 Turbo	CoT	2 Shot	0.87	0.88	0.85	0.89
GPT4 Turbo	CoT	4 Shot	0.89	0.90	0.92	0.91
GPT4 Turbo	CoT	8 Shot	0.93	0.92	0.94	0.94
GPT4 Turbo	CoT **[E]**	8 Shot	**0.94**	**0.93**	**0.96**	**0.95**
Claude 2.1	Regular	Zero Shot	0.76	0.77	0.71	0.62
Claude 2.1	Regular	4 Shot	0.87	0.84	0.87	0.85
Claude 2.1	CoT	2 Shot	0.84	0.81	0.83	0.84
Claude 2.1	CoT	4 Shot	0.91	0.84	0.90	0.94
Claude 2.1	CoT	8 Shot	0.90	0.84	0.90	0.94
Claude 2.1	CoT **[E] [*]**	8 Shot	***0.91***	***0.86***	***0.91***	***0.95***
Mixtral	Regular	Zero Shot	0.71	0.66	0.45	0.51
Mixtral	Regular	4 Shot	0.81	0.82	0.79	0.75
Mixtral	CoT	2 Shot	0.79	0.75	0.63	0.70
Mixtral	CoT	4 Shot	0.84	0.86	0.77	0.76
Mixtral	CoT	8 Shot	0.89	0.86	0.80	0.81
Mixtral	CoT **[E]**	8 Shot	0.89	0.88	0.81	0.82

Table 2: Comparing the results generated by the models with different prompting strategies. **[CoT]** - denotes chain-of-thought. **[E]** - denotes Ensemble (as described in 4.4). **[*]** - denotes submission during the test phase on the Leaderboard.

deduction process.

Our experiments revealed promising results. While zero-shot performance was limited, providing explanatory prompts substantially improved the models' reasoning abilities. Performance increased with more prompt specificity and steps. Our best results came from an ensemble approach applying majority voting across multiple chain-of-thought prompts.

Ultimately, our system achieved competitive rankings on the leaderboard, placing 2^{nd} in the word puzzle sub-task and 13^{th} on sentence puzzles. The strong capability unlocked through guided prompting highlights these models' latent reasoning potential when given a structured thought process. Our work sheds light on how explanatory chains can elicit more of the knowledge within large language model parameters.

A few key limitations remain to be addressed in future work. First, constructing effective prompts requires extensive human effort and insight - automating this prompting process could improve scalability. Additionally, performance still lags behind human levels, indicating that there is room for advancement. Architectural constraints related to long-term memory and reasoning likely need to be overcome. Finally, our approach focused narrowly on the given puzzles rather than teaching broader inferential skills - developing more generalizable reasoning through prompts is an open challenge.

Acknowledgments

We would like to thank the shared task organizers for proposing this interesting competition and for providing participants with the BrainTeaser dataset.

References

Anthropic. 2023. Claude 2.1: Updates and improvements. https://www.anthropic.com/news/claude-2-1.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*.

Dhiman Goswami, Md Nishat Raihan, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023. OffMix-3L: A novel code-mixed test dataset in bangla-english-hindi for offensive language identification. In *Proceedings of SocialNLP (AACL)*.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, and Hai-Tao Zheng. 2023. Lateval: An interactive llms evaluation benchmark with incomplete information from lateral thinking puzzles. *arXiv preprint arXiv:2308.10855*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024a. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024b. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of EMNLP*.

Peng Liu, Nicholas Lourie, Sean Welleck, and Julia Hockenmaier. 2021. What makes good in-context examples for gpt-3? In *Proceedings of EMNLP*.

Shentong Mo and Miao Xin. 2023. Tree of uncertain thoughts reasoning for large language models. *arXiv preprint arXiv:2309.07694*.

OpenAI. 2023. Gpt-4 technical report. https://arxiv.org/abs/2303.08774.

Md Nishat Raihan, Dhiman Goswami, Antara Mahmud, Antonios Anastasopoulos, and Marcos Zampieri. 2023a. SentMix-3L: A novel code-mixed test dataset in bangla-english-hindi for sentiment analysis. In *Proceedings of SEALP (AACL)*.

Md Nishat Raihan, Umma Hani Tanmoy, Anika Binte Islam, et al. 2023b. Offensive language identification in transliterated and code-mixed bangla. In *Proceedings of BLP (EMNLP)*.

Sirwe Saeedi, Aliakbar Panahi, Seyran Saeedi, and Alvis C Fong. 2020. Cs-nlp team at semeval-2020 task 4: Evaluation of state-of-the-art nlp deep learning architectures on commonsense reasoning task. *arXiv preprint arXiv:2006.01205*.

Juanhe TJ Tan. 2023. Causal abstraction for chain-of-thought reasoning in arithmetic word problems. In *Proceedings of BlackboxNLP (EMNLP)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, et al. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Kankan Zhou, Eason Lai, Wei Bin Au Yeong, Kyriakos Mouratidis, and Jing Jiang. 2023. Rome: Evaluating pre-trained vision-language models on reasoning beyond visual common sense. *arXiv preprint arXiv:2310.19301*.

Yanyan Zou and Wei Lu. 2019. Joint detection and location of English puns. In *Proceedings of NAACL*.

MasonTigers at SemEval-2024 Task 8: Performance Analysis of Transformer-based Models on Machine-Generated Text Detection

Sadiya Sayara Chowdhury Puspo*, **Md Nishat Raihan***, **Dhiman Goswami***,
Al Nahian Bin Emran, Amrita Ganguly, Özlem Uzuner
George Mason University, USA
spuspo@gmu.edu

Abstract

This paper presents the *MasonTigers'* entry to the SemEval-2024 Task 8 - Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection. The task encompasses Binary Human-Written vs. Machine-Generated Text Classification (Track A), Multi-Way Machine-Generated Text Classification (Track B), and Human-Machine Mixed Text Detection (Track C). Our best performing approaches utilize mainly the ensemble of discriminator transformer models along with sentence transformer and statistical machine learning approaches in specific cases. Moreover, zero-shot prompting and fine-tuning of FLAN-T5 are used for Track A and B.

1 Introduction

In academia and beyond, machine-generated content is proliferating across news platforms, social media, forums, educational materials, and scholarly works. Breakthroughs in large language models (LLMs), like GPT-3.5 and GPT-4, facilitate the creation of fluent responses to diverse user queries. While this capability raises prospects of replacing human labor in various tasks, concerns arise about potential misuse, including the generation of deceptive misinformation (Chen and Shu, 2023) and completing student assignments, which hinders the development of essential writing skills (Jungherr, 2023).This highlights the importance of developing automated systems to detect and mitigate the potential abuse of machine-generated content, as well as distinguishing between machine-written and human-generated text. Additionally, Prior studies (ZeroGPT [1]; Mitchell et al., 2023; Bao et al., 2023) predominantly adopted a binary classification approach for machine-generated text (MGT), with a primary focus on English. However, there has been limited research addressing the amalgamation of human-written and MGT texts (Wang et al., 2024d).

In response to these limitations, SemEval-2024 introduces a shared task: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection (Wang et al., 2024c). This task comprises three subtasks, each targeting different aspects of machine-generated text complexity.Subtask A focuses on Binary Human-Written vs. MGT Classification, involving two tracks: monolingual and multilingual. Subtask B tackles Multi-Way Machine-Generated Text Classification to identify the source of a given text. Subtask C involves detecting the transition point within a mixed text, determining where it shifts from human-written to machine-generated. The data provided for this task is an expansion of the M4 dataset (Wang et al., 2024d) and benchmark evaluation of (Wang et al., 2024b). .

In conducting these tasks, we conduct a range of experiments and observe that ensemble methods outperform individual models significantly in classification tasks, e.g., Goswami et al. (2023) Emran et al. (2024), Ganguly et al. (2024). Our weighted ensemble approaches achieve accuracies of 74%, 60% and 65% in subtask A monolingual; multilingual tracks and subtask B respectively, given that we have used different models for both tasks. In subtask C, we explore different setups, ensembling which results in Mean Absolute Error (MAE) of 60.78. For the classifications, we utilize zero-shot prompting and fine-tuning of FlanT5[2], while adhering to the restriction of no data augmentation in this task.

2 Related Works

The difficulties of separating human-written text from large language models and the significance of

*** denotes equal contribution.**

[1] www.zerogpt.com/

[2] huggingface.co/google/flan-t5-base/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1364–1372

Source	Train					Dev	
	chatGPT	Cohere	Davinci	Dolly	Human	Bloomz	Human
arxiv	3000	3000	2999	3000	15498	500	500
peerread	2344	2342	2342	2344	2357	500	500
reddit	3000	3000	3000	3000	15500	500	500
wikihow	3000	3000	3000	3000	15499	500	500
wikipedia	2995	2336	3000	2702	14497	500	500
Total				**54406**	**63351**	**2500**	**2500**

Table 1: Label Distribution of Train and Validation Data for Binary Human-Written vs. Machine-Generated Text Classification (Subtask A - Monolingual)

trustworthy methods for evaluation are highlighted by recent research (e.g. Chaka 2024, Elkhatat et al. 2023). In terms of human evaluation of MGT, Guo et al. (2023) indicates that generated texts from large language models tend to exhibit less emotional and objective content compared to human-written texts. Tang et al. (2023) suggests that distinct signals left in the machine-generated text may facilitate the identification of suitable features to differentiate between MGT and human-written texts. Whereas, Sadasivan et al. (2023) observes that detection techniques become less effective as language models improve. Moreover, Ippolito et al. (2019) advocates for the importance of using both human and automatic detectors to assess the humanness of text generation systems.

Previous work in determining MGT from human-written ones include higher order n-grams (Gallé et al., 2021), utilizing linguistic patterns (Muñoz-Ortiz et al., 2023), curvature-based criterion (Mitchell et al., 2023), tweaking with multiple variables (Dugan et al., 2023), fine-tuning transformer-based models e.g., Capobianco; Chen and Liu (2023). Very recently, Wang et al. (2024a) puts forward LLM-Detector, offering a fresh method for identifying text at both document and sentence levels by employing Instruction Tuning of LLMs. To tackle challenges of this field, several datasets have been released, e.g., MULTITuDE (Macko et al., 2023), M4 (Wang et al., 2024d). Additionally, there have been multiple shared tasks organized related to this topic (Shamardina et al., 2022a; Molla et al., Molla et al.; Kashnitsky et al., 2022. Despite several collective findings and techniques, as argued by Sadasivan et al. (2023), there remains a critical need for the creation of reliable detection methods capable of accurately distinguishing between human and machine-generated text, a requirement essential across both English and other languages.

3 Datasets

Wang et al. (2024d) collects datasets from a variety of sources, including Wikipedia (the March 2022 version), WikiHow (Koupaee and Wang, 2018), Reddit (ELI5), arXiv, PeerRead (Kang et al., 2018)(for English), and Baike (for Chinese). They employ web question answering for Chinese, news content for Urdu, Indonesian, and RuATD (Shamardina et al., 2022b) for Russian language. The method of prompting machine-generated text (MGT) has been extensively outlined in Wang et al. (2024d).

Subtask A, Binary Human-Written vs. Machine-Generated Text Classification, in the monolingual track involves a same-domain cross-generator experiment, where instances are exclusively in English and gathered from five distinct sources with two labels: 0 and 1. Human-generated texts receive a label of 0, while machine-generated texts from four different LLMs (chatGPT, Cohere, *davinci-003*, and Dolly-v2) are labeled as 1. The distribution of Train and Validation datasets, both in terms of labels and sources, along with the number of test instances, is detailed in Tables 1. During the test phase, there are 16,272 instances labeled as 0 and 18,000 instances labeled as 1.

On the other hand, Subtask A in the multilingual track entails a cross-domain same-generator experiment. Instances are sourced from nine different sources during the training phase, including four different languages, while the validation dataset comprises three different languages as indicated in Table 2. Similar to the monolingual task, human-generated texts are labeled as 0, and machine-generated texts from five different LLMs (Bloomz (Muennighoff et al., 2022), chatGPT, Cohere, *davinci-003*, and Dolly-v2) are labeled as 1. In the test phase, there are 20,238 instances labeled as 0 and 22,140 instances labeled as 1.

Source	Train						Dev		
	Bloomz	**chatGPT**	**Cohere**	**Davinci**	**Dolly**	**Human**	**ChatGPT**	**Davinci**	**Human**
arxiv	3000	3000	3000	2999	3000	15998	-	-	-
peerread	2334	2344	2342	2344	2344	2857	-	-	-
reddit	2999	3000	3000	3000	3000	16000	-	-	-
wikihow	3000	3000	3000	3000	3000	15999	-	-	-
wikipedia	2999	2995	2336	3000	2702	14997	-	-	-
Bulgarian	0	3000	0	3000	0	6000	-	-	-
Chinese	0	2970	0	2964	0	6000	-	-	-
Indonesian	0	3000	0	0	0	2995	-	-	-
Urdu	0	2899	0	0	0	3000	-	-	-
Arabic	-	-	-	-	-	-	500	0	500
German	-	-	-	-	-	-	500	0	500
Russian	-	-	-	-	-	-	500	500	1000
Total					**83571**	**83846**		**2000**	**2000**

Table 2: Label Distribution of Train and Validation Data for Binary Human-Written vs. Machine-Generated Text Classification (Subtask A - Multilingual)

Source	Train						Dev					
	Bloomz	**chatGPT**	**Cohere**	**Davinci**	**Dolly**	**Human**	**Bloomz**	**chatGPT**	**Cohere**	**Davinci**	**Dolly**	**Human**
arxiv	3000	3000	3000	2999	3000	2998	-	-	-	-	-	-
reddit	2999	3000	3000	3000	3000	3000	-	-	-	-	-	-
wikihow	3000	3000	3000	3000	3000	2999	-	-	-	-	-	-
wikipedia	2999	2995	2336	3000	2702	3000	-	-	-	-	-	-
peerread	-	-	-	-	-	-	500	500	500	500	500	500
Total	**11998**	**11995**	**11336**	**11999**	**11702**	**11997**	**500**	**500**	**500**	**500**	**500**	**500**

Table 3: Label Distribution of Train and Validation Data for Multi-Way Machine-Generated Text Classification (Subtask B)

Label	**Test Data**
Human (0)	3000
chatGPT (1)	3000
cohere (2)	3000
davinci (3)	3000
Bloomz (4)	3000
Dolly (5)	3000
Total	18000

Table 4: Label Distribution of Test Data for Multi-Way Machine-Generated Text Classification (Subtask B)

Subtask B, Multi-Way Machine-Generated Text Classification, represents another cross-domain same-generator experiment. In contrast to Subtask A, Subtask C involves six labels: 0 for human, 1 for chatGPT, 2 for Cohere, 3 for *davinci-003*, 4 for Bloomz, and 5 for Dolly. These labels correspond to instances sourced from five different sources. However, it's noteworthy that the sources for the training and validation data differ, and this distinction is outlined in Tables 3 and 4.

Subtask C, involving Human-Machine Mixed Text Detection, provides a composite text with a human-written first part followed by a machine-generated second part. The task is to discern the boundary, and labels are provided as word indices to distinguish it. The label distribution of data is shown in Table 5.

Data	**Count**
Train	3649
Dev	505
Test	11123

Table 5: Number of Instances for Human-Machine Mixed Text Detection (Subtask C)

4 Experimental Setup

4.1 Data Preprocessing

In the monolingual track of subtask A, we received approximately 160K instances for training and development. To preserve the text's integrity, we eliminate special characters, extra new lines, unnecessary whitespace, and hyperlinks from the data, ensuring that only the essential text remains in sub-

tassk A (monolingual), B & C. However, in the multilingual track of subtask A, since none of our team members are familiar with the languages present in the instances, we only remove hyperlinks. We ensure that punctuation marks such as full stops, commas, and exclamation signs are retained in all instances, as they play a crucial role in this task (Tang et al., 2023).

4.2 Hyperparameters

In our experimental setup, we configure several key parameters to train our model effectively. We utilize a batch size of 16, controlling the number of training samples processed in each iteration, learning being set to 1e-5, and incorporating dropout with a rate of 0.25 to prevent overfitting by randomly dropping a fraction of units during training. Maintaining a fixed sequence length of 512 tokens ensured consistency in input data processing. For optimization, we employ the AdamW optimizer (Loshchilov and Hutter, 2017), known for its efficacy in training deep neural networks with added weight decay regularization. These experiments are conducted on a 80GB NVIDIA A100 GPU machine over the period of 24 hours, leveraging its computational power and memory capacity. By systematically adjusting these parameters, we aim to understand their influence on the model's performance, ultimately optimizing our approach for the task at hand. The adjustment of these parameters is carried out in both subtask A & B.

4.3 Models: SubTask A

In monolingual track, we employ Roberta (Liu et al., 2019), DistilBERT (Sanh et al., 2019), and ELECTRA (Clark et al., 2020). Subsequently, we apply a weighted ensemble method, incorporating RoBERTa, DistilBERT, and ELECTRA, employing a voting strategy due to their closely comparable individual accuracies. The weights are their corresponding accuracy.

Similarly, in the multilingual track, we utilize LASER (Li and Mak, 2020), mBERT (Devlin et al., 2018), and XLMR (Goyal et al., 2021). Following that, we deploy a weighted ensembled strategy involving these models, utilizing the voting method.

4.4 Models: SubTask B

Subtask B, poses a considerable challenge, as opposed to the first two tracks where the model distinguishes between human and machine-generated text. Here, the model must differentiate among human-generated text and five distinct LLMs. For this, we leverage Roberta, ELECTRA, Deberta (He et al., 2020), and subsequently create a weighted (weights are set as acauracy) ensemble approach of these models using voting technique.

4.5 Models: SubTask C

In subtask C, we find the embedding of the training data using Term Frequency - Inverse Document Frequency (TF-IDF) (Aizawa, 2003), Positive Point-wise Mutual Information (PPMI) (Church and Hanks, 1990), and the embedding using language model RoBERTa (Liu et al., 2019). Then for each training embedding generated by these approaches, we apply Linear Regression (Groß, 2003) and ElasticNet (Zou and Hastie, 2005) separately on these embeddings and predict the first word or index of from where the machine-generated text started in a specific data instance. We selected the word that is the starting word of the closest neighboring paragraph as the predicted word index. Then we clip the predicted values to ensure the predictions range from 0 to the length of the specific data instance (rounded if necessary). In the development phase, we find the Mean Absolute Error (Chai and Draxler, 2014) of these six predictions (three each by Linear Regression and ElasticNet). Then we perform a weighted ensemble depending on the Mean Absolute Error of the six predicted results and get our ensembled MAE in the development phase. We also perform this approach on the test data and find our smallest MAE in the evaluation phase.

4.6 Prompting and Fine-Tuning LLM

> **Prompt:** <*Text*> If this piece of text is Human Generated, answer 0 or If Machine Generated, answer 1.

Figure 1: Sample FlanT5 prompt.

For subtasks A & B, we experiment with FlanT5 zero-shot prompting, utilizing the Hugging Face Transformers[3] library, specifically the T5ForConditionalGeneration class and T5Tokenizer. Training is conducted on an NVIDIA A100 GPU with 80GB memory over 24 hours. The prompting sample for subtask A is shown in Figure 1. In subtask B, we maintain consistency in prompting by keeping the question the same as

[3] huggingface.co/docs/transformers/

Monolingual			Multilingual		
Model	**Dev**	**Test**	**Model**	**Dev**	**Test**
Baseline (RoBERTa)	0.74	0.88	**Baseline (XLM-R)**	0.72	0.81
FLAN-T5 Prompting	0.49	0.52	FLAN-T5 Prompting	0.42	0.39
FLAN-T5 Fine-tuning	0.57	0.53	FLAN-T5 Fine-tuning	0.48	0.43
RoBERTa	0.70	0.73	LASER	0.52	0.50
DistilBERT	0.69	0.70	mBERT	0.57	0.58
ELECTRA	0.78	0.71	XLMR	0.61	0.59
Ensemble (Wt. accuracy)	0.79	0.74	**Ensemble (Wt. accuracy)**	0.63	0.60

Table 6: Accuracy of Binary Human-Written vs. Machine-Generated Text Classification (Subtask A)

labeling the human-generated text as "1", while prompting the machine-generated texts from various Language Model Models (LLMs) as categories "2" through "6."

We also finetune a t5-small model over 2 epochs, setting the learning rate to 0.001 and the batch size to 4. We employ a full finetuning (FFT) approach without the utilization of any quantization method like LoRa (Hu et al., 2021) or QLoRA (Dettmers et al., 2023). Due to the adoption of an FFT approach and the sheer size of the dataset, we do not experiment with a wide set of hyper-parameters. We empirically choose a few combinations and report the best results.

5 Results

Subtask A and B are evaluated based on Accuracy, as specified by (Wang et al., 2024c), while Subtask C employs Mean Absolute Error (MAE) as the evaluation metric [4].

In the monolingual track of Subtask A, ELECTRA demonstrates superior accuracy (0.78) compared to RoBERTa (0.70) and DistilBERT (0.69) during the development phase. Consequently, the weighted ensemble of these three models achieves an accuracy of 0.79 in our development submission, surpassing the baseline RoBERTa model. Upon publishing test labels, a comparison with the test label results reveals accuracies detailed in Table 6, with the ensemble model achieving an accuracy of 0.74, while the baseline accuracy increases to 0.88, differing by 0.14 compared to the development phase. In the multilingual track, XLM-R outperforms LASER and mBERT with an accuracy of 0.61. Ensembling these models achieves accuracies of 0.63 in the development phase and 0.60 in the test phase, whereas the baseline accuracies are 0.72 and 0.81, respectively. Both zero-shot prompting and fine-tuning FlanT5 demonstrate less than satisfactory performance, yielding accuracies of 0.53 and 0.43 in the monolingual and multilingual tracks, respectively.

Model	**Dev**	**Test**
Baseline (RoBERTa)	0.75	0.75
FLAN-T5 Prompting	0.54	0.48
FLAN-T5 Fine-tuning	0.57	0.54
RoBERTa	0.72	0.56
ELECTRA	0.73	0.59
DeBERTa	0.77	0.64
Ensemble (Wt. accuracy)	0.79	0.65

Table 7: Accuracy of Multi-Way Machine-Generated Text Classification (Subtask B)

Within subtask B, DeBERTa outperforms RoBERTa and ELECTRA, achieving superior performance with an accuracy of 0.77. Ensembling these models yields accuracies of 0.79 and 0.65 in both the development and test phases, whereas baseline RoBERTa gives 0.75 in both phases. Similar to subtask A, fine-tuning and prompting FLAN T5 exhibit suboptimal results in both phases shown in Table 7.

In subtask C, various methods are considered, and it is found that RoBERTa with Elastic Net achieved the minimum Mean Absolute Error (33.28). Table 8 highlights that Elastic Net outperforms Linear Regression in terms of lower MAE during both the development and test phases. To enhance predictive performance, we employ a weighted ensemble of development phase MAE of six combinations, resulting in MAE values of 31.71 and 60.78 during the development and test phases, respectively. However, the baseline (longformer) model gives MAE of 3.53 ± 0.21 and 21.54.

[4] https://github.com/mbzuai-nlp/SemEval2024-task8

Model	Dev	Test
Baseline (Longformer)	$\simeq 3.53$	21.54
TF-IDF + LR	44.15	71.23
PPMI + LR	41.93	68.41
RoBERTa + LR	37.52	65.82
TF-IDF + EN	38.36	67.09
PPMI + EN	35.67	63.36
RoBERTa + EN	33.28	62.34
Wt. (dev. MAE) Ensemble	31.71	60.78

Table 8: Mean Absolute Error(MAE) value of Human-Machine Mixed Text Detection (Subtask C) (LR = Linear Regression, EN = ElasticNet)

6 Error Analysis

In the monolingual track of Subtask A, the final model demonstrates proficiency in accurately identifying machine-generated text. Nonetheless, there is a notable presence of false positives, indicating instances where the model incorrectly identifies human-written texts as machine-generated. Despite this, the model effectively detects machine-generated text without omission. Similarly, in the multilingual track of Subtask A, the ultimate model excels in accurately distinguishing machine-generated text. However, false positives are prevalent, indicating numerous cases where human-written texts are inaccurately classified as machine-generated. Additionally, the model encounters instances where it fails to predict machine-generated texts.

In Subtask B, the model excels in accurately predicting chatGPT-generated texts. However, its performance declines notably for davinci-generated text, often misclassifying it as chatGPT generated. Additionally, the model's accuracy is lower for Dolly-generated and human-written texts, indicating a discrepancy in handling machine-generated versus human-written content.

For subtask C, MAE is higher due to the presence of outliers because the dev MAE was significantly lower than the test MAE. To handle this issue, it is essential to address the preprocessing of data, handling outliers, selecting appropriate features, optimizing model complexity, improving data quality, and ensuring model stability through proper tuning and evaluation procedures. This can be the future scope of research in this specific domain.

For a clearer understanding, refer to the visual evaluations in Figure 2, 3, 4 of Appendix.

7 Conclusion

In our investigation of SemEval-2024 Task 8, we applied a diverse set of methodologies, encompassing statistical machine learning techniques, transformer-based models, sentence transformers, and FLAN T5. Subtask A involved binary classification, where the monolingual track focused on cross-generator scenarios within the same domain, and the multilingual track addressed cross-domain scenarios within the same generators. Subtask B dealt with multi-label classification, requiring the discrimination of human-generated text from five distinct language models. Subtask C centered on Human-Machine Mixed Text Detection, employing TF-IDF, PPMI, and RoBERTa with Linear Regression and ElasticNet for prediction. The outcomes of three subtasks highlighted the efficacy of ensemble methods, showcasing specific models excelling in each subtask. Additionally, we explored the applicability of zero-shot prompting and fine-tuning FLAN-T5 for Tracks A and B.

In summary, our approach harnessed a blend of transformer models, machine learning methodologies, and ensemble strategies to tackle the complexities presented by SemEval-2024 Task 8. The paper underscores the imperative need for robust detection methods to effectively navigate the growing prevalence of machine-generated content.

Limitations

The task involved extensive datasets in each phase of all subtasks, leading to prolonged execution times and increased GPU usage. Additionally, the texts themselves were lengthy. Moreover, the prohibition of additional data augmentation added to the complexity of the task. The nuanced distinction between human-written and machine-generated text, which can sometimes be challenging for humans to discern, poses an even greater difficulty for models attempting to learn this differentiation.

Acknowledgements

We express our gratitude to the organizers for orchestrating this task and to the individuals who diligently annotated datasets across various languages. Your dedication has played a crucial role in the triumph of this undertaking.

References

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Marc Capobianco. *Supervised Machine Generated Text Detection Using LLM Encoders In Various Data Resource Scenarios*. Ph.D. thesis, WORCESTER POLYTECHNIC INSTITUTE.

Tianfeng Chai and Roland R Draxler. 2014. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*.

Chaka Chaka. 2024. Reviewing the performance of ai detection tools in differentiating between ai-generated and human-written texts: A literature and integrative hybrid review. *Journal of Applied Learning and Teaching*.

Canyu Chen and Kai Shu. 2023. Combating misinformation in the age of llms: Opportunities and challenges. *arXiv preprint arXiv:2311.05656*.

Zheng Chen and Huming Liu. 2023. Stadee: Statistics-based deep detection of machine generated text. In *Proceedings of ICIC*, pages 732–743. Springer.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of AAAI*.

Ahmed M Elkhatat, Khaled Elsaid, and Saeed Almeer. 2023. Evaluating the efficacy of ai content detection tools in differentiating between human and ai-generated text. *International Journal for Educational Integrity*.

Al Nahian Bin Emran, Amrita Ganguly, Sadiya Sayara Chowdhury Puspo, Dhiman Goswami, and Md Nishat Raihan. 2024. Masonperplexity at climateactivism 2024: Integrating advanced ensemble techniques and data augmentation for climate activism stance and hate event identification. *arXiv preprint arXiv:2402.01976*.

Matthias Gallé, Jos Rozen, Germán Kruszewski, and Hady Elsahar. 2021. Unsupervised and distributional detection of machine-generated text.

Amrita Ganguly, Al Nahian Bin Emran, Sadiya Sayara Chowdhury Puspo, Md Nishat Raihan, Dhiman Goswami, and Marcos Zampieri. 2024. Masonperplexity at multimodal hate speech event detection 2024: Hate speech and target detection using transformer ensembles. *arXiv preprint arXiv:2402.01967*.

Dhiman Goswami, Md Nishat Raihan, Sadiya Sayara Chowdhury Puspo, and Marcos Zampieri. 2023. nlpbdpatriots at blp-2023 task 2: A transfer learning approach to bangla sentiment analysis. In *Proceedings of BLP (EMNLP)*.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. In *Proceedings of RepL4NLP*.

Jürgen Groß. 2003. *Linear regression*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Andreas Jungherr. 2023. Using chatgpt and other large language model (llm) applications for academic paper assignments.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine Van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications. *arXiv preprint arXiv:1804.09635*.

Yury Kashnitsky, Drahomira Herrmannova, Anita de Waard, Georgios Tsatsaronis, Catriona Fennell, and Cyril Labbé. 2022. Overview of the dagpap22

shared task on detecting automatically generated scientific papers. In *Proceedings of SDP*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Wei Li and Brian Mak. 2020. Transformer based multilingual document embedding model. *arXiv preprint arXiv:2008.08567*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Samuel Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, et al. 2023. Multitude: Large-scale multilingual machine-generated text detection benchmark. *arXiv preprint arXiv:2310.13606*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

Diego Molla, Haolan Zhan, Xuanli He, and Qiongkai Xu. 2023. Overview of the 2023 alta shared task: Discriminate between human-written and machine-generated text. *Training*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Alberto Muñoz-Ortiz, Carlos Gómez-Rodríguez, and David Vilares. 2023. Contrasting linguistic patterns in human and llm-generated text. *arXiv preprint arXiv:2308.09067*.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected?

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022a. Findings of the the ruatd shared task 2022 on artificial text detection in russian. In *Computational Linguistics and Intellectual Technologies*.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022b. Findings of the the ruatd shared task 2022 on artificial text detection in russian. *arXiv preprint arXiv:2206.01583*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts.

Rongsheng Wang, Haoming Chen, Ruizhe Zhou, Han Ma, Yaofei Duan, Yanlan Kang, Songhua Yang, Baoyu Fan, and Tao Tan. 2024a. Llm-detector: Improving ai-generated chinese text detection with open-source llm instruction tuning. *arXiv preprint arXiv:2402.01158*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. 2024b. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. *arXiv preprint arXiv:2402.11175*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024c. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024d. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of EACL*.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*.

A Appendix

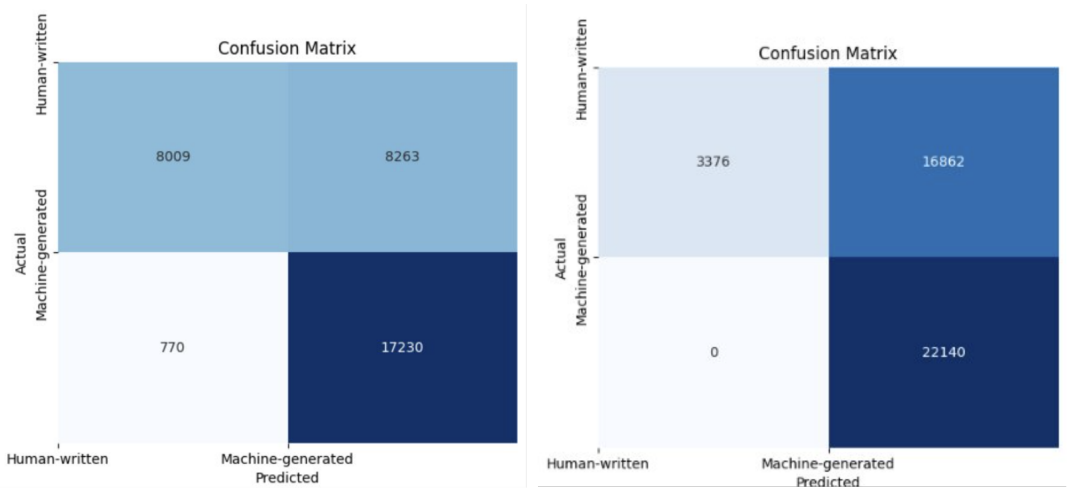


Figure 2: Confusion Matrix (Binary Human-Written vs. Machine-Generated Text Classification : Monolingual (Left), Multilingual (Right))

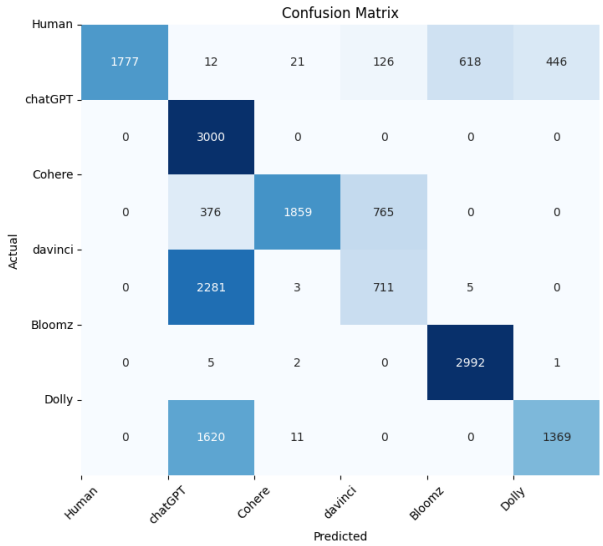


Figure 3: Confusion Matrix (Multi-Way Machine-Generated Text Classification)

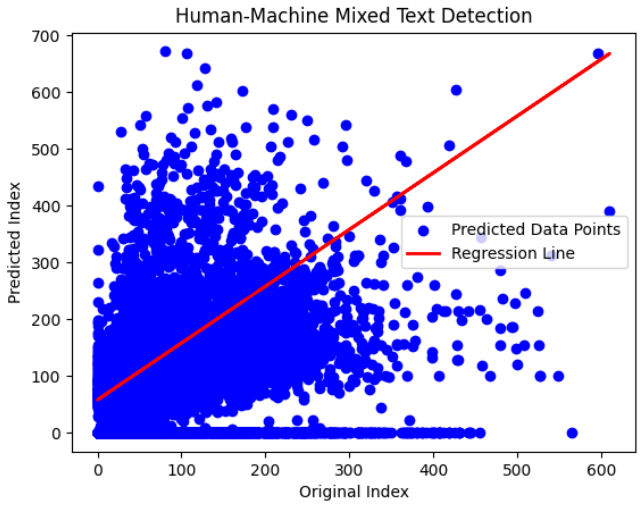


Figure 4: Regression (Human-Machine Mixed Text Detection)

UIC NLP GRADS at SemEval-2024 Task 3: Two-Step Disjoint Modeling for Emotion-Cause Pair Extraction

Sharad Chandakacherla*, **Vaibhav Bhargava***, and **Natalie Parde**
Department of Computer Science
University of Illinois at Chicago, USA
{schand65, vbharg4, parde}@uic.edu

Abstract

Disentangling underlying factors contributing to the expression of emotion in multimodal data is challenging but may accelerate progress toward many real-world applications. In this paper we describe our approach for solving SemEval-2024 Task #3, Sub-Task #1, focused on identifying utterance-level emotions and their causes using the text available from the multimodal *F.R.I.E.N.D.S.* television series dataset. We propose to disjointly model emotion detection and causal span detection, borrowing a paradigm popular in question answering (QA) to train our model. Through our experiments we find that (a) contextual utterances before and after the target utterance play a crucial role in emotion classification; and (b) once the emotion is established, detecting the causal spans resulting in that emotion using our QA-based technique yields promising results.

1 Introduction

The task of **e**motion **c**ause **a**nalysis in **c**onversations (Wang et al., 2023, ECAC) aims to decipher the expression of human emotion in conversational data, either through unimodal (text-only) or multimodal (e.g., with the addition of video and/or audio) information. On a fundamental level, this is a complex two-part problem: emotion must be identified for a given utterance, and the span of dialogue causing that emotion must subsequently be recognized.[1] SemEval-2024 Task #3 (Wang et al., 2024) was organized around solving this problem, broken into two subtasks varying in the data allowed to build the solution; in Sub-Task #1, identification of emotion cause was limited to the use of only text information. We address Sub-Task #1 in this paper.

We pursued two strategies in our approach toward solving the task. First, we trained a question answering (QA) model (Rajpurkar et al., 2018) to extract causal spans given the reference emotions for non-neutral utterances within the training set. In doing so, we achieved comparable results to those reported by Wang et al. (2023) and Poria et al. (2021), the latter of which is a popular benchmark for this task. Next, we devised a two-step disjoint model that separately learns to classify emotion and extract causal spans during training. During inference we (1) run the emotion classifier, enriching the test set with emotion labels; and (2) run inference on the QA model to extract the causal spans. Our approach achieved third place according to the primary task metric (a weighted-average proportional F_1) and 2nd place on the secondary metric (weighted-average strict F_1; see §A.2 for results on all relevant task metrics). We elaborate on our findings in the remainder of this paper.[2]

2 Background

2.1 Task Description

Given a conversation with a sequence of n emotional utterances $u \in \{u_1, ..., u_n\}$, the twin goals in SemEval Task #3 are to identify (a) the emotion label $e_i \in \{$HAPPINESS, SADNESS, DISGUST, FEAR, SURPRISE, ANGER, NEUTRAL$\}$; and (b) for emotions other than those identified as NEUTRAL, the corresponding span of text c_i that caused u_i to be assigned label e_i.

Input and Output. Each input u_i is a sequence of text. This text is matched with video and audio in the dataset, although for Sub-Task #1 only the text is used for learning and inference. Output for each u_i is a categorical label e_i in the label space defined previously, and a sequence of text c_i signifying the reason why e_i was assigned to u_i.

*Authors contributed equally.

[1] *Neutral* utterances have no corresponding causal spans.

[2] Our source code is publicly available at: https://github.com/sharadchandakacherla/MultiModalEmotionCauseAnalysis/tree/main/submission.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1373–1379

Attribute	Frequency
# Conversations	1374
# Utterances	13619
anger	11.85%
disgust	3.03%
fear	2.74%
joy	16.90%
sadness	8.42%
surprise	13.51%
neutral	43.53%

Table 1: Dataset statistics. *# Conversations* and *# Utterances* are raw frequency counts, whereas all emotion categories are percentages of total utterances.

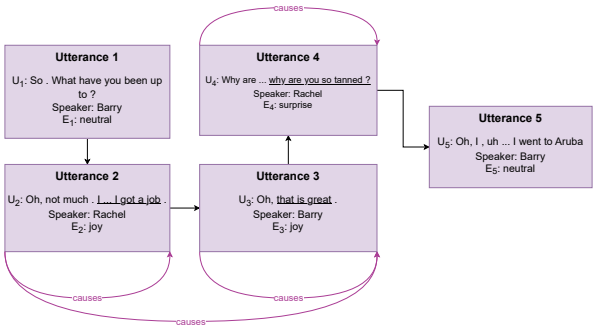


Figure 1: An example conversation from the dataset.

Dataset. All Task #3 entries were trained and evaluated using Wang et al. (2023)'s multi-modal conversational emotion cause dataset (ECF). ECF is an English-language dataset sourced from transcripts, audio, and video clips from the popular television sitcom *F.R.I.E.N.D.S*; the series comprises daily informal conversations involving a cast of six friends living in New York City. Conversations are segmented into individual speaker utterances, referred to as "emotional utterances." Causal spans are linked to emotional utterances in the dataset, and annotators could source them from any utterance in the given conversation. Dataset statistics, including the distribution of emotion labels across utterances in ECF, are presented in Table 1. Sample inputs to the emotion classification model and the causal span extractor are shown in Figure 1.

2.2 Related Work

ECAC has been studied previously to some extent in both disjoint and joint training settings. ECE (Gui et al., 2016) introduced a dataset with emotion causes extracted from a Chinese news article corpus; the language in this dataset is formal and in passive voice. Instances place focus on both clause-level (to capture emotion) and phrase-level (to capture boundaries) annotations. Building on this, ECPE (Xia and Ding, 2019) proposed a joint training model to extract emotion and corresponding causal spans, using word2vec embeddings (Mikolov et al., 2013) pre-trained on a corpus extracted from a Chinese micro-blogging website. They used a two-step process to address the emotion-cause pair extraction task, performing emotion extraction and cause extraction first, followed by emotion-cause pairing and filtering using a hierarchical Bi-LSTM model.

Poria et al. (2019) introduced a novel multimodal, multi-party conversational dataset for emotion recognition in conversations (MELD). Wang et al. (2023) makes use of MELD, and created another corpus of emotional utterances paired with their causes; this corpus also serves as the dataset for our task. Wang et al. (2023) establish baselines for the Multimodal Emotion-Cause Pair Extraction in Conversations (MC-ECPE) task using the same guidelines described in ECPE. The authors of RECCON (Poria et al., 2021) solve the task of recognizing emotion cause in conversations using causal span extraction and causal emotion entailment with transformer-based models. However, they employ their methods on IEMOCAP (Busso et al., 2008) which is a dyadic dataset and DailyDialog (Li et al., 2017) which consists of manually written dialogues focusing on physically-situated topics. Other prior work has used formal conversation datasets or reported speech where emotions are often expressed explicitly (Gui et al., 2016). Performing emotion classification and causal span extraction using a QA-based paradigm for an informal conversational setting is a novel approach to link emotion causes to implicitly expressed emotion.

3 System Overview

We model the task to maximize the probability of finding emotion-cause pairs, (e_i, c_i), for the given conversation context x. We disjointly train models with parameters θ and ϕ to estimate the emotion e_i from x and the causal span c_i from (e_i, x), respectively. We approximate x to the prompts x_e and x_c to provide the appropriate contextual information to our models, as shown in Equation 1.

$$\begin{aligned} P_{\theta,\phi}(e_i, c_i|x) &= P_\theta(e_i|x)P_\phi(c_i|e_i, x) \\ &\approx P_\theta(e_i|x_e)P_\phi(c_i|e_i, x_c) \end{aligned} \quad (1)$$

Our approach is a two-step process, by which we first identify e_i for the given utterance u_i in a

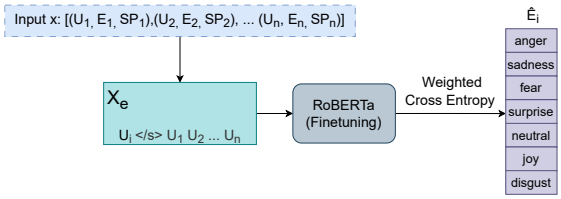


Figure 2: Training the *emotion classifier*.

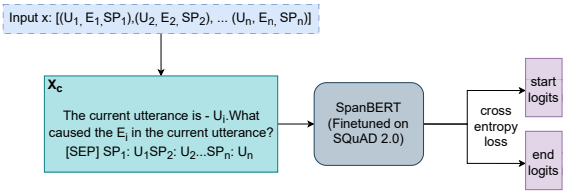


Figure 3: Training the *emotion causal classifier*.

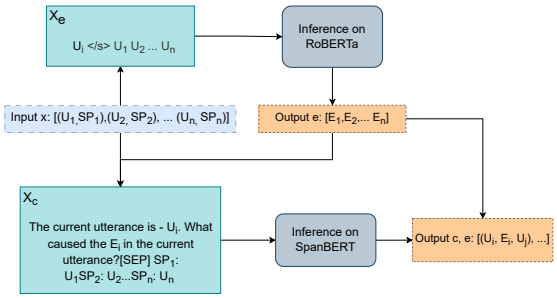


Figure 4: Performing inference at test time.

conversation from x_e, and then identify the causal span c_i for u_i in all cases when $e_i \neq$ NEUTRAL. We fine-tune separate pre-trained language models (PLMs) for these two steps. While fine-tuning for emotion cause spans, we use the emotion labels provided as part of the training set to construct x_c.

Emotion Classifier. We use a RoBERTa base model (Zhuang et al., 2021) as the backbone of our emotion classification model, with a weighted cross entropy loss to penalize emotion label predictions. We use class weights from our training set as weighing terms for the loss function, and fine-tune for 20 epochs using the AdamW optimizer (Loshchilov and Hutter, 2019) with linear rate scheduler with 500 warm-up steps. We use the learning rate 5×10^{-5} with 0.01 weight decay rate, and select the best epoch based on weighted F_1. The input prompt to this model is a space-separated concatenation of u_i, the separator token proposed by Zhuang et al. (2021), and all utterances in the conversation ($U_{\text{all}} = \{u_1, ..., u_n\}$) space-separated in sequential order, as shown in Figure 2.

Emotion Cause Classifier. We frame emotion cause classification as a QA task. To avoid asking our model to answer impossible questions, we skip utterances where the predicted label is NEUTRAL. We use a SpanBERT base model (Joshi et al., 2020) fine-tuned on SQuAD2.0 (Rajpurkar et al., 2018).[3] We then further fine-tune this model on our task. The input prompt to this model is:

> `The current utterance is - [`u_i`]. What caused the` e_i `in the current utterance? <SEP>` U_{all}

This is shown in Figure 3. For fine-tuning SpanBERT, we change the batch size from 32 to 12 and the maximum sequence length from 512 to 400. We set the learning rate to 2×10^{-5} and train the model for five epochs. Figure 4 shows inference using our proposed system. We first perform inference on our *emotion classifier* for the test dataset to augment test x_c with emotion labels e_i, and then perform inference on our *emotion cause classifier*.

4 Experimental Setup

ECF is already split into *train* and *test* sets. We separate a *dev* set from *train* by holding out the last 20% of the training data. We make use of pre-trained RoBERTa (Liu et al., 2019) and SpanBERT (Joshi et al., 2020) models from HuggingFace. Other details regarding our hardware and software libraries can be found in §A.3.

For training the *emotion classifier* we make use of weighted F_1 score, choosing the best performing model based on this metric. While training the *emotion cause classifier*, we select models based on metrics defined by Joshi et al. (2020) for span-based learning with PLMs: unweighted exact match, and F_1 score.

5 Results

5.1 Main Quantitative Findings

Our proposed system achieves 3rd place in SemEval Task #3, Sub-Task #1, based on the primary task metric of weighted average proportional F_1 (Wang et al., 2023). We achieve 2nd place overall based on the auxiliary metric of weighted average strict F_1,[4] which accounts for exact span matching. We show

[3] We observe that this additional fine-tuning boosts our model's performance (Appendix A.1).

[4] https://github.com/NUSTM/SemEval-2024_ECAC/tree/main/CodaLab/evaluation

Metric	Score	Ranking
w-avg. Strict F_1	0.1839	2
w-avg. Proportional F_1	**0.2442**	**3**
Strict F_1	0.1851	2
Proportional F_1	0.2397	4

Table 2: Official task scores, shown alongside final leaderboard rankings for Sub-Task #1.

Model	Metric	Score
Our Model	**w-avg. Strict F_1**	**0.2741**
Wang et al.	w-avg. Strict F_1	0.2625

Table 3: Comparing our model's performance on the *dev* set to Wang et al. (2023)'s text-only baseline.

our *test* scores for all official task metrics in Table 2. In Table 3 we compare to Wang et al. (2023)'s baseline for this task, showing that our model improves upon this baseline. Results reported in Table 3 are based on *dev* performance, since *test* was held private by the task organizers.

5.2 Quantitative Analysis

To investigate the performance of our approach, we used the *dev* dataset to perform an ablation study regarding prompt context length and fine-tuning data for the *emotion cause classifier*. We also experimented with varied scaling factors and input context for *emotion classification*.

5.2.1 Emotion Classification

Scaling Factors. We experimented with the use of class size as a scaling factor to improve performance for less-represented classes (e.g., *disgust* or *fear*). In Table 4, models post-fixed with *(w)* are scaled versions of the emotion classification model trained with a weighted cross-entropy loss. We observe large performance differences for under-represented classes under this condition; however, we also observe a slightly reduced F_1 overall. This is a positive observation for our disjoint training regime, as the causal span extractor model isn't trained on neutral cases during training, and it confirms the utility of class weight scaling for this task.

Input Context. We also experimented with varied input context, adjusting the context of the input by passing only `u_i` compared to the longer `u_i <SEP> U_all` used in our final model.

	u_i	u_i (w)	u_i </s> U_{all}	u_i </s> $U_{\text{all}}(w)$
Anger	0.46	0.45	**0.48**	**0.48**
Disgust	0.10	0.23	**0.24**	0.20
Fear	0.17	0.26	0.20	**0.27**
Joy	0.55	0.53	0.59	**0.60**
Sadness	**0.76**	0.74	0.73	0.72
Surprise	0.38	0.35	0.39	**0.40**
Neutral	0.63	0.53	**0.64**	0.58
F_1	0.60	0.58	0.60	0.59

Table 4: Ablation study on different prompts for the *emotion classifier*. </s> is the special token and (w) represents models trained with a weighted cross-entropy loss. F_1 is weighted average strict F_1.

5.2.2 Emotion Cause Classification

We experimented with two QA configurations examining prompt context length and fine-tuning data, shown in Table 5. In the former, we tweaked the model's maximum sequence length for models using the complete set of utterances in a conversation, U_{all}. We compared our results to a model trained only on prior context, i.e., $u \in \{u_1, ..., u_i\}$ where u_i is the current utterance. Interestingly, such models exhibited slightly higher F_1 scores; this is comparable to causal span extraction scores in (Poria et al., 2021). In the latter, we compared versions of our approach using (a) the pretrained SpanBERT directly, and (b) a version that was fine-tuned using SQuAD 2.0 data. We observed that additional training on a model previously trained on the SQuAD 2.0 dataset yields better performance than the pretrained SpanBERT model.

Sequence Length	EM	F_1
400	**0.4466**	**0.6133**
512	0.4397	0.6095
Model		
SpanBERT & SQuAD 2.0	**0.5147**	**0.6810**
SpanBERT	0.4428	0.6494

Table 5: Ablation study on sequence length (full context) and model for the *emotion causal classifier*. *EM* is exact match, and *F_1* is weighted average strict F_1. The base model of SpanBERT used is `spanbert-base-cased`. We prompt the model with only past and current context.

5.3 Error Analysis

We analyzed mispredicted examples from the *dev* set to identify commonly occurring errors that

might not be readily apparent by the w-avg proportional F1-score, and we observed that some of these conversations had neutral utterances with no corresponding emotion-cause pairs. From the 275 conversations, there were 23 such instances of which 12 were composed of only neutral utterances. In such cases, our span extractor model's output is accurate as it simply skips such utterances by design, and when neutral utterances are predicted correctly, this is the correct action. Conversely, in the cases where there are emotional utterances yet no causal pairs provided, the span model is unpredictable as it is not trained to predict empty causal spans, reinforcing our hypotheses grounded in Equation 1, i.e., that results of span extraction are dependent on the emotion classification model.

We also observed errors where incorrect spans were predicted for correctly identified emotions. In many instances, this involved the prediction of causal spans from *future* utterances. Given the nature of the data (informal conversations), it is possible for future utterances to overlap temporally with the current utterance. In other cases, it might seem to a third-person viewer that the cause of an emotion expressed at a timestep *t* becomes apparent after an utterance from a later timestep. Our model was not able to handle such cases predictably. Following manual review of all 32 predictions made for causal spans appearing in future utterances, we found that only 7 predictions were correct, mostly for the numerous emotion classes. This supports our rationale behind fine-tuning both our models in a full-context setting as explained earlier (§3), but suggests that there is still room for improvement. We suspect that the autoregressive context studied in follow-up analyses (§A.1) may result in better performance by skipping such examples, or perhaps a jointly-trained or multimodal model would help bridge the shortcomings.

Finally, we present a sample of correct predictions and mispredictions in Table 6. We observe that emotion classification for the most representative classes works as hypothesized, i.e., the emotions *joy* and *surprise* are predicted better than *fear*. For the span extraction task, we observe that rows 3 and 4 with non-neutral emotion predictions have "N/A" as their span prediction as, in these instances, the best prediction for an utterance with multiple causes returned identical spans as rows 2 and 5, respectively. One way to avoid such cases could be to pair all utterances u_i and u_j along with u_{all} ($u_i, u_j \in \{u_1, ..., u_n\}$), resulting in a quadratic increase in resource requirements and clipped inputs due to the model's limited token length. However, as this behavior was not consistent across all cases, we opted for the simpler solution described in §3. This also helped with resource constraints.

Utterance	Gold Emo.	Pred. Emo.	Gold Cause	Pred. Cause
Thank you. Oh Joey and look at this crib! It is so cute!	joy	joy	look at this crib! It is so cute!	It is so cute !
I know! I found it on the street.	joy	joy	It is so cute!	look at this crib! It is so cute!
I know! I found it on the street.	joy	joy	I found it on the street.	N/A
Are you serious ... Really ?! It is in such good condition.	surprise	surprise	I found it on the street.	N/A
Are you serious ... Really ?! It is in such good condition.	surprise	surprise	It is in such good condition.	It is in such good condition.
Yeah.	joy	neutral	It is in such good condition.	N/A
Wow! Whoa ... whoa what under the covers?	surprise	surprise	what under the covers?	It is in such good condition.
Ew.	fear	disgust	It is moving.	It is moving.
It is still ... It is got a tail! Get it out of here! Get it out of here!!	fear	fear	It is got a tail!	It is moving .
Ooh! Ah! Okay!	fear	surprise	It is got a tail!	It is moving.

Table 6: Correct and incorrect predictions from *dev*.

6 Conclusion

We introduce a disjoint model comprising an emotion classifier and an emotion-cause classifier. Our system addresses emotion cause extraction competitively based on the official leaderboard and on follow-up analyses (§5). We set out with the objectives of developing a disjoint model making use of

the entire conversation to identify emotions in utterances and using a well-known QA paradigm to extract the causes of the emotions, and we achieve this with varying degrees of success. We observe that emotion classification is harder than emotion cause extraction when emotion annotations are present (Tables 4 and 5), and that when the model assigns emotions correctly, it also has a greater chance of extracting causal spans correctly (Table 6). This is more evident when only prior contexts are present, yielding higher scores.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback, and the SemEval-2024 Task 3 organizers for introducing us to this challenging and intriguing research problem.

References

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Ebrahim (Abe) Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. 2008. Iemocap: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42:335–359.

Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, and Yu Zhou. 2016. Event-driven emotion cause extraction with corpus construction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1639–1649, Austin, Texas. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, Alexander Gelbukh, and Rada Mihalcea. 2021. Recognizing emotion cause in conversations.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Appendix

A.1 SQuAD 2.0 Fine-Tuning Affects Emotion Cause Classification

We observe a strong increase in performance of the *emotion cause classifier* if additional fine-tuning is performed using SQuAD 2.0 (Rajpurkar et al., 2018). In this case, the model is prompted with The

current utterance is u_i What caused the e_i in current utterance?. We did not consider utterances $u \in \{u_{i+1}, ..., u_n\}$. The unweighted exact match and F_1 increases, as shown in Table 5.

A.2 Other Metrics for the Model

Metric	Value
Weighted strict precision	0.339
Weighted strict recall	0.235
Weighted strict F-1	0.274
Weighted Proportional precision	0.425
Weighted Proportional recall	0.288
Weighted Proportional F-1	**0.339**
Strict precision	0.348
Strict recall	0.235
Strict F-1	0.280
Proportional precision	0.431
Proportional recall	0.280
Proportional F-1	0.339

Table 7: Additional results for our model on the *dev* set as defined by Wang et al. (2023). Weighted Proportional F_1 was the primary metric used for SemEval Task #3.

We provide additional results for other metrics defined by Wang et al. (2023) in Table 7.

A.3 Hardware and Hyperparameters

We make use of PyTorch 2.2.0,[5] HuggingFace transformers 4.37.2 for the RoBERTa-base implementation,[6] FAIR's[7] spanbert-base-cased, FAIR's SpanBERT fine-tuned on SQuAD2.0 and sklearn 1.3.2[8] to fine-tune our models. We train our models using an Nvidia RTX 3090Ti GPU.

[5] https://pytorch.org/get-started/locally/

[6] https://huggingface.co/docs/transformers/en/installation

[7] https://github.com/facebookresearch/SpanBERT/

[8] https://scikit-learn.org/stable/install.html

MasonTigers at SemEval-2024 Task 1: An Ensemble Approach for Semantic Textual Relatedness

Dhiman Goswami, Sadiya Sayara Chowdhury Puspo, Md Nishat Raihan, Al Nahian Bin Emran, Amrita Ganguly, Marcos Zampieri
George Mason University, USA
dgoswam@gmu.edu

Abstract

This paper presents the *MasonTigers*' entry to the SemEval-2024 Task 1 - Semantic Textual Relatedness. The task encompasses supervised (Track A), unsupervised (Track B), and cross-lingual (Track C) approaches to semantic textual relatedness across 14 languages. *MasonTigers* stands out as one of the two teams who participated in all languages across the three tracks. Our approaches achieved rankings ranging from 11^{th} to 21^{st} in Track A, from 1^{st} to 8^{th} in Track B, and from 5^{th} to 12^{th} in Track C. Adhering to the task-specific constraints, our best performing approaches utilize an ensemble of statistical machine learning approaches combined with language-specific BERT based models and sentence transformers.

1 Introduction

In this modern era of information retrieval and NLP, understanding semantic relatedness is fundamental for refining and optimizing diverse applications. Semantic relatedness refers to the degree of similarity and cohesion (Hasan and Halliday, 1976) in meaning between two words, phrases, or sentences. Semantic relatedness allows systems to grasp the contextual and conceptual connections between words or expressions. Various NLP tasks and applications can benefit from modeling semantic relatedness such as question answering (Park et al., 2014), knowledge transfer (Rohrbach et al., 2010), text summarization (Rahman and Borah, 2023), machine translation (Ali et al., 2009), and content recommendation (Piao et al., 2016).

While significant research has been conducted on semantic relatedness in English, more recently the interest in semantic relatedness in other languages has been steadily growing (Baldissin et al., 2022). This reflects an increasing awareness of the need for developing models to languages English other than English. NLP is evolving rapidly and we have been witnessing the emergence of language-specific transformer, the release of datasets for downstream tasks in diverse languages, and the development of multilingual models designed to handle linguistic diversity.

SemEval-2024 Task 1 - Semantic Textual Relatedness (Ousidhoum et al., 2024b) aims to determine the semantic textual relatedness (STR) of sentence pairs across 14 diverse languages. Track A focuses on nine languages (Algerian Arabic, Amharic, English, Hausa, Kinyarwanda, Marathi, Moroccan Arabic, Spanish, Telugu) using a supervised approach where systems are trained on labeled training datasets. Track B adopts an unsupervised approach, prohibiting the use of labeled data to indicate similarity between text units exceeding two words. This track encompasses 12 languages (Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Modern Standard Arabic, Moroccan Arabic Punjabi, and Spanish). Track C involves cross-lingual analysis across the 12 aforementioned languages. Participants in this track must utilize labeled training data from another track for at least one language, excluding the target language. Evaluation across all three tracks involves using Spearman Correlation between predicted similarity scores and human-annotated gold scores. We conduct distinct experiments for each track using statistical machine learning approaches along with the embeddings generated by transformer based models.

2 Related Work

Understanding the level of semantic relatedness between two languages has been regarded as essential for grasping their meaning. Notable studies on the topic including Agirre et al. (2012, 2013, 2014, 2015, 2016); Dolan and Brockett (2005) and Li et al. (2006) have introduced datasets like STS, MRPC, and LiSent. These datasets have been piv-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1380–1390

otal in advancing research in tasks such as text summarization and plagiarism detection.

Finding semantic relatedness and semantic similarity, as well as determining sentence pair similarity using existing datasets or paired annotation, are integral in understanding the nuances of language comprehension. Previous studies describe how words and sentences are perceived to convey similar meanings (Halliday and Hasan, 2014; Morris and Hirst, 1991; Asaadi et al., 2019; Abdalla et al., 2021; Goswami et al., 2024).

Methodologies like paired comparison represent the most straightforward type of comparative annotations (Thurstone, 1994), (David, 1963). Best-Worst Scaling (BWS) (Louviere and Woodworth, 1991) a comparative annotation schema, offer insights into methods for evaluating relatedness through pairwise comparisons. The utilization of these methods aids in generating ordinal rankings of items based on their semantic relatedness. Kiritchenko and Mohammad (2016, 2017) highlight the effectiveness of such techniques, emphasizing the importance of reliable scoring mechanisms derived from comparative annotations for understanding the intricacies of semantic relatedness in natural language processing tasks.

3 Data

The shared task comprises three tracks: Supervised, Unsupervised, and Cross-Lingual. The dataset (Ousidhoum et al., 2024a) is comprised of two columns: the initial column, labeled "text," containing two full sentences separated by a special character, and the second column, labeled as "score", which includes degree of semantic textual relatedness for the corresponding pair of sentences. In the supervised track (Track A), there are 9 languages, and for each language, train, dev, and test sets are provided. The specifics of the dataset for this track can be found in Table 1.

Language	Train	Dev	Test
Algerian Arabic (arq)	1,261	97	583
Amharic (amh)	992	95	171
English (eng)	5,500	250	2,600
Hausa (hau)	1,736	212	603
Kinyarwanda (kin)	778	102	222
Marathi (mar)	1,200	293	298
Moroccan Arabic (ary)	924	71	426
Spanish (esp)	1,562	140	600
Telugu (tel)	1,170	130	297

Table 1: Track A Dataset Distribution

In the unsupervised track (Track B), there are 12 languages and for all the languages dev and test set is provided. The details of the dataset of this track is available in Table 2.

Language	Dev	Test
Afrikaans (afr)	20	375
Algerian Arabic (arq)	97	583
Amharic (amh)	95	171
English (eng)	250	2,600
Hausa (hau)	212	603
Hindi (hin)	288	968
Indonesian (ind)	144	360
Kinyarwanda (kin)	102	222
Modern Standard Arabic (arb)	32	595
Moroccan Arabic (ary)	71	426
Punjabi (pan)	242	634
Spanish (esp)	140	600

Table 2: Track B Dataset Distribution

Finally, in the cross-lingual track (Track C), there are 12 languages and for all the languages dev and test set is provided and they are same as the unsupervised track. Here the training dataset is not provided. Hence, for each individual language of this track, we select 5 languages from supervised track (different from the target language) and merge training data of those five languages to create the training dataset for each of the languages of cross-lingual track. The details of the dataset of this track is available in Table 3.

4 Experiments

We use statistical machine learning along with language specific BERT-based models to find the sentence embeddings and predict relatedness between pair of sentences. Additionally, we use sentence transformers for the supervised track. Our experiments are described in detail in the next sections.

4.1 Track A - Supervised

At first, we find the embedding of the training data using Term Frequency - Inverse Document Frequency (TF-IDF) (Aizawa, 2003), Positive Pointwise Mutual Information (PPMI) (Church and Hanks, 1990), and Language-Agnostic BERT Sentence Embedding (LaBSE sentence transformer) (Feng et al., 2020) separately. Also we find the embeddings using language specific BERT based models. For Algerian Arabic, Amharic, English, Hausa, Kinyarwanda, Marathi, Moroccan Arabic, Spanish and Telugu - DziriBERT (Abdaoui et al., 2021),

Language	Train Data from (Track A)	Train	Dev	Test
Afrikaans (afr)	amh, eng, esp, arq, ary	10,239	20	375
Algerian Arabic (arq)	amh, hau, esp, eng, ary	10,714	97	583
Amharic (amh)	eng, hau, esp, arq, ary	10,983	95	171
English (eng)	arq, ary, mar, esp, tel	6,117	250	2,600
Hausa (hau)	amh, esp, arq, ary, eng	10,239	212	603
Hindi (hin)	esp, eng, mar, ary, tel	10,356	288	968
Indonesian (ind)	ary, eng, mar, esp, tel	5,356	144	360
Kinyarwanda (kin)	amh, esp, ary, arq, eng	10,239	102	222
Modern Standard Arabic (arb)	amh, eng, arq, esp, ary	10,239	32	595
Moroccan Arabic (ary)	amh, hau, eng, esp, arq	11,051	71	426
Punjabi (pan)	arq, esp, mar, eng, tel	10,693	242	634
Spanish (esp)	arq, ary, mar, eng, tel	10,055	140	600

Table 3: Track C Data Distribution (Train Data from Track A)

AmRoBERTa (Yimam et al., 2021), RoBERTa (Liu et al., 2019), HauRoBERTa (Adelani et al., 2022), KinyaBERT (Nzeyimana and Niyongabo Rubungo, 2022), MarathiBERT (Joshi, 2022b), DarijaBERT (Gaanoun et al., 2024), SpanishBERT (Cañete et al., 2020) and TeluguBERT (Joshi, 2022a) are used.

For each training embedding, we calculate the cosine similarity (Rahutomo et al., 2012) between the pairs. After that we apply ElasticNet (Zou and Hastie, 2005) and Linear Regression (Groß, 2003) separately on these embeddings and predict the relatedness of the sentence pairs in the development phase. We clip the predicted values to ensure the prediction range from 0 to 1. In the development phase, we find the Spearman Correlation Coefficient (Myers and Sirois, 2004) of these eight predictions (four each by ElasticNet and Linear Regression). Finally, we perform a weighted ensemble depending on the Spearman Correlation Coefficient of the eight predicted results and get our ensembled Spearman Correlation Coefficient in development phase. We also perform this approach on the test data and find our best Spearman Correlation Coefficient in the evaluation phase.

4.2 Track B - Unsupervised

For unsupervised track, we find the embedding of the development data using Term Frequency - Inverse Document Frequency (TF-IDF) (Aizawa, 2003) and Positive Point-wise Mutual Information (PPMI) (Church and Hanks, 1990) separately. Also we find the embeddings using language specific BERT based models. For Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Modern Standard Arabic, Moroccan Arabic, Punjabi and Spanish - AfricanBERTa, DziriBERT (Abdaoui et al., 2021), AmRoBERTa (Yimam et al., 2021), RoBERTa (Liu et al., 2019), HauRoBERTa (Adelani et al., 2022), HindiBERT (Joshi, 2022a), IndoBERT (Koto et al., 2020), KinyaBERT (Nzeyimana and Niyongabo Rubungo, 2022), ArabicBERT (Safaya et al., 2020), DarijaBERT (Gaanoun et al., 2024), PunjabiBERT (Joshi, 2022a), and SpanishBERT (Cañete et al., 2020) are used.

Then for each development embedding generated by these three approaches, we calculate cosine similarity (Rahutomo et al., 2012) between the pairs. In the development phase, we find the Spearman correlation (Myers and Sirois, 2004) of these values calculated on embeddings found by three different procedures and perform an average ensemble of the calculated results to get our ensembled Spearman correlation in development phase. We also perform this approach on the test data and find our best Spearman correlation in the evaluation phase.

4.3 Track C - Cross-Lingual

For each language in cross-lingual track, we select 5 different languages from Supervised Track to use as training data. The details of the language selection is provided in Table 3. The we find the embedding of the training data using Term Frequency - Inverse Document Frequency (TF-IDF) (Aizawa, 2003) and Positive Point-wise Mutual Information (PPMI) (Church and Hanks, 1990) separately. Also we find the embeddings using language specific (unrelated to the target language) BERT based models. For Afrikaans, Amharic, Hausa and Kinyarwanda - we use ArabicBERT (Safaya et al., 2020), for Algerian Arabic, Modern Standard Arabic and Moroccan Arabic - we

use AfricanBERTa[1], for English, Hindi, Indonesian, Punjabi and Spanish - SpanishBERT (Cañete et al., 2020), BanglaBERT (Bhattacharjee et al., 2022), RoBERTa-tagalog (Cruz and Cheng, 2021), HindiBERT (Joshi, 2022a) and RoBERTa (Liu et al., 2019) are used. Then for each training embedding generated by these three approaches, we calculate cosine similarity (Rahutomo et al., 2012) between the pairs. After that we apply ElasticNet (Zou and Hastie, 2005) and Linear Regression (Groß, 2003) separately on these embeddings and predict the similarity of the sentence pairs in the development phase. We clip the predicted values to ensure the prediction range from 0 to 1. In the development phase, we find the Spearman correlation of these six predictions (three each by ElasticNet and Linear Regression) and perform an average ensemble of the predictions to get our ensembled Spearman correlation in development phase. We also perform this approach on the test data and find our best Spearman correlation in the evaluation phase.

5 Results

For all the tracks, ensemble of the predictions prove helpful in terms of achieving better Spearman correlation.

For Track A sentence transformer LaBSE along with Linear Regression performs the best among the eight combinations for all the languages. Then the weighted ensemble improves the result 1% - to 3% in development phase and 1% - 2% in evaluation phase - depending on the languages. For English this method performs the best in terms of ranking with 11^{th} rank while the worst for Moroccan Arabic with 21^{th} rank. On test Spearman correlation, English is the best securing 0.84 and Kinyarwanda is the worst with 0.37. Detailed results are shown in Table 4 of Appendix.

For Track B, embedding generated by language specific BERT based models provide the best result among the three combinations for all the languages. Then the average ensemble improves the result 0% - to 3% in development phase and 0% - 2% in evaluation phase - depending on the languages. For Kinyarwanda this method performs the best in terms of ranking with 1^{st} rank while the worst for English with 8^{th} rank. On test Spearman correlation, English is the best securing 0.77 and Punjabi is the worst with 0.02. Detailed result is shown in Table 5 of Appendix.

For Track C embedding generated by language specific (unrelated to target language) BERT based models provide the best result among the six combinations for all the languages. Then the average ensemble improves the result 0% - to 2% in both development and evaluation phases depending on the languages. For Punjabi this method performs the best in terms of ranking with 5^{th} rank while the worst for Hausa and Kinyarwanda with 12^{th} rank. On test Spearman correlation, Spanish is the best securing 0.56 and Punjabi is the worst with 0.02. Detailed result is shown in Table 6 of Appendix.

6 Error Analysis

For Track A, Algerian Arabic, Moroccan Arabic and Spanish test Spearman Correlation Coefficient decreases in the evaluation phase. This happens because the dev set was around 7.5%-9% and the test set is around 39% - 46% size of the train set.

For Track B, amount of dev data was only 20 for Afrikaans which is the reason of a very big difference between the result of development and evaluation phase. Algerian Arabic, Amharic, Modern Standard Arabic, Moroccan Arabic have a very small amount of dev data (less than 100) which is reason of decreased Spearman Correlation Coefficient in the evaluation phase. Hindi also faces the same issue but as it had more dev data the test Spearman Correlation Coefficient is only 4% less than the development period.

For Track C, Algerian Arabic, Indonesian, Kinyarwanda, Modern Standard Arabic faced bigger drop of the Spearman Correlation Coefficient from the development phases. The main issue here is the BERT based models that doesn't know the target languages generate the embeddings that are not as good as what we observed in unsupervised track for the models with the knowledge of target language. Also the diversity of the train and test data make it more challenging to score better Spearman Correlation Coefficient. In addition, due to the unavailability of the text label, only the ensemble performance of Spanish language for all the tracks are shown.

Regarding the result of the Punjabi language in the both unsupervised and cross-lingual track, it was the most challenging language where the provide baseline was less than zero. Though our system achieves 0.02 Spearman Correlation Coefficient for for this language, the ranking is quite

[1] https://huggingface.co/mrm8488/AfricanBERTa

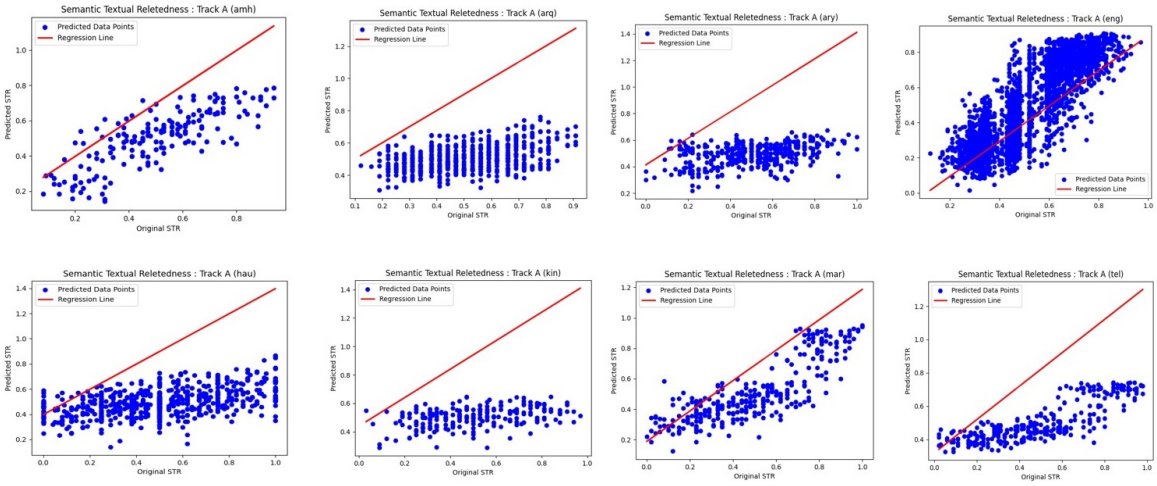


Figure 1: Track A (Comparison with gold semantic textual relatedness)

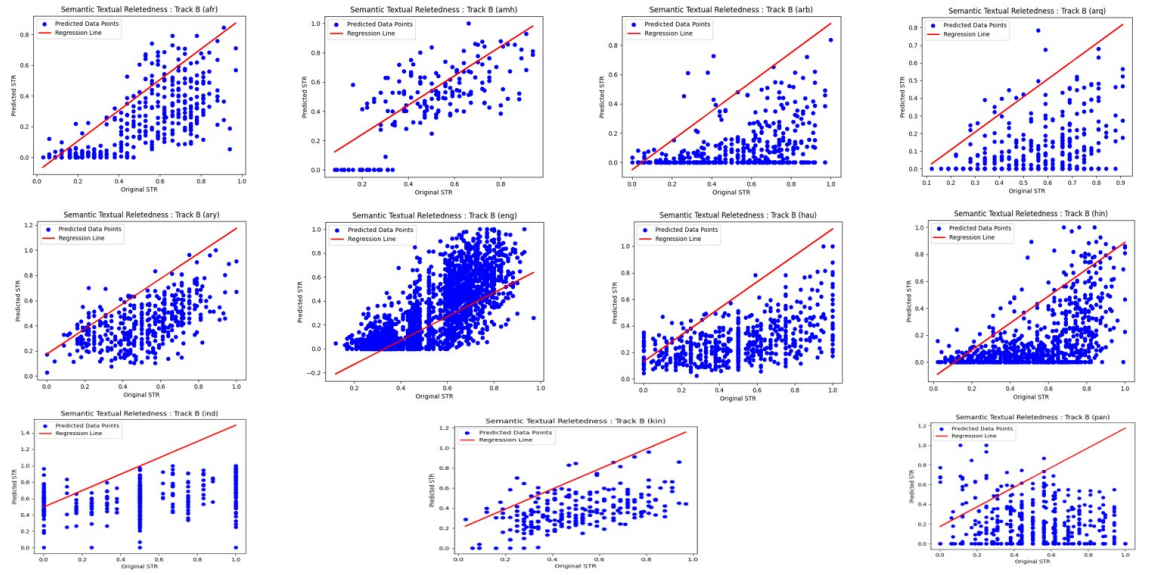


Figure 2: Track B (Comparison with gold semantic textual relatedness)

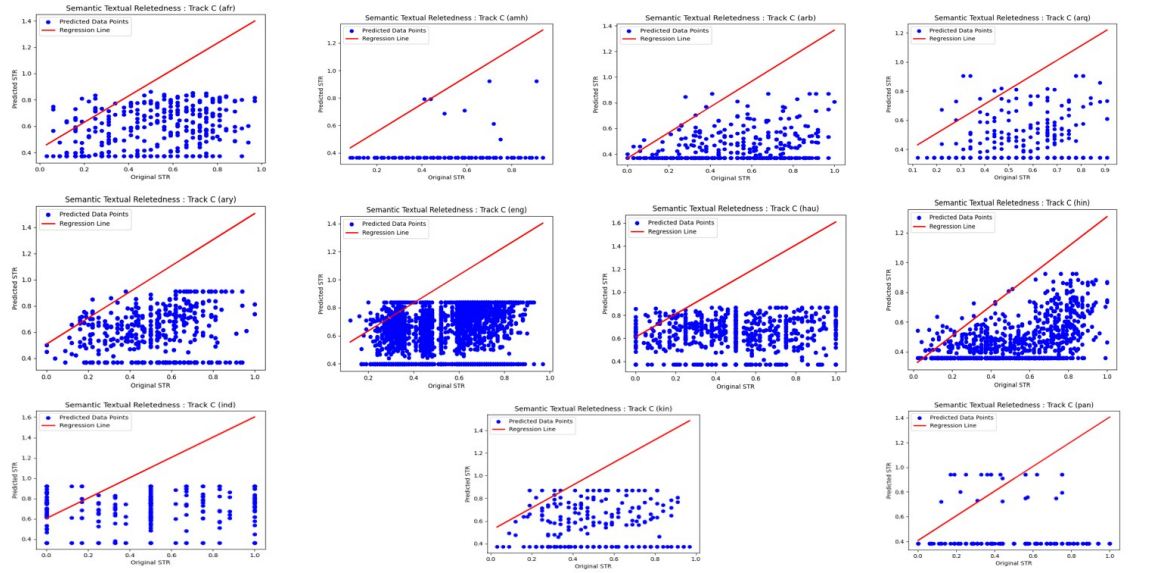


Figure 3: Track C (Comparison with gold semantic textual relatedness)

impressive which also proves the struggle of other teams to cope up with this language.

Moreover, ElasticNet and Linear Regression exhibit limitations as assumption of linearity may not align with the intricate and nonlinear relationships inherent in the textual data. The issue of dimensionality poses a challenge, especially when dealing with a large number of features. The difference between the gold and predicted semantic relatedness scores for the three tracks are shown in Figure 1, Figure 2, and Figure 3.

7 Conclusion

We experimented with various methodologies on the dataset provided by the organizers, including statistical machine learning approaches, transformer-based models, language-specific BERTs, and sentence BERT. In the supervised task (Track A), with no restrictions on the model or data, we utilized the available training dataset. Conversely, the unsupervised task (Track B), lacking training data, presented challenges, leading us to use language-specific BERTs and statistical machine learning approaches. The cross-lingual track (Track C) imposed more stringent restrictions, requiring us to use training data from other languages in Track A, excluding the target language. In addition to statistical ML models, we integrated language-specific BERTs closely aligned with the geography and culture of the target language, as the use of LLMs was constrained due to unknown training data.

We show that our ensemble approach exhibited superior performance compared to individual model experiments. However, the task's inherent difficulty became evident in instances where relatively small datasets presented challenges for effective model learning. Semantic textual relatedness tasks face challenges like subjectivity, context dependency, and ambiguity due to multiple meanings and cultural differences. Limited data, domain specificity, short texts, and biases hinder accuracy. Ongoing research is crucial to address these limitations and improve model accuracy.

Acknowledgements

We would like to thank the shared task organizers for providing participants with the dataset used in this paper.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M Mohammad. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *arXiv preprint arXiv:2110.04845*.

Amine Abdaoui, Mohamed Berrimi, Mourad Oussalah, and Abdelouahab Moussaoui. 2021. Dziribert: a pre-trained language model for the algerian dialect. *arXiv preprint arXiv:2109.12346*.

David Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba Alabi, Shamsuddeen Muhammad, Peter Nabende, et al. 2022. Masakhaner 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of EMNLP*.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of SemEval*.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of SemEval*.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of SemEval*.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of SemEval*.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Proceedings of * SEM*.

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*.

Ola Mohammad Ali, Mahmoud GadAlla, and Mohammad Said Abdelwahab. 2009. Improving word sense disambiguation in machine translation using semantic relatedness and statistical measures of association. In *Proceedings of ICICIS*.

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. 2019. Big bird: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In *Proceedings of NAACL*.

Gioia Baldissin, Dominik Schlechtweg, and Sabine Schulte im Walde. 2022. Diawug: A dataset for diatopic lexical semantic variation in spanish. In *Proceedings of LREC*.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of NAACL*.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *Proccedings of PML4DC (ICLR)*.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*.

Jan Christian Blaise Cruz and Charibeth Cheng. 2021. Improving large-scale language models and resources for filipino. *arXiv preprint arXiv:2111.06053*.

Herbert Aron David. 1963. *The method of paired comparisons*. London.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Procedings of IWP)*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Kamel Gaanoun, Abdou Mohamed Naira, Anass Allak, and Imade Benelallam. 2024. Darijabert: a step forward in nlp for the written moroccan dialect. *International Journal of Data Science and Analytics*.

Dhiman Goswami, Sadiya Sayara Chowdhury Puspo, Md Nishat Raihan, and Al Nahian Bin Emran. 2024. Masontigers@ LT-EDI-2024: An ensemble approach towards detecting homophobia and transphobia in social media comments. *arXiv preprint arXiv:2401.14681*.

Jürgen Groß. 2003. *Linear regression*. Springer Science & Business Media.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. 2014. *Cohesion in english*. Routledge.

Ruqaiya Hasan and Michael AK Halliday. 1976. Cohesion in english. *London, 1976; Martin JR*.

Raviraj Joshi. 2022a. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Raviraj Joshi. 2022b. L3Cube-MahaCorpus and MahaBERT: Marathi monolingual corpus, Marathi BERT language models, and resources. In *Proceedings of WILDRE*.

Svetlana Kiritchenko and Saif M Mohammad. 2016. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of HLT (NAACL)*.

Svetlana Kiritchenko and Saif M Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. *arXiv preprint arXiv:1712.01765*.

Fajri Koto, Afshin Rahimi, Jey Han Lau, and Timothy Baldwin. 2020. Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian nlp. In *Proceedings of COLING*.

Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jordan J Louviere and George G Woodworth. 1991. Best-worst scaling: A model for the largest difference judgments. Technical report, Working paper.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics*.

Leann Myers and Maria J Sirois. 2004. Spearman correlation coefficients, differences between. *Encyclopedia of statistical sciences*.

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. KinyaBERT: a morphology-aware Kinyarwanda language model. In *Proceedings of ACL*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Seonyeong Park, Hyosup Shim, and Gary Geunbae Lee. 2014. Isoft at qald-4: Semantic similarity-based question answering system over linked data. In *CLEF (Working Notes)*.

Guangyuan Piao, Safina Showkat Ara, and John G Breslin. 2016. Computing the semantic similarity of resources in dbpedia for recommendation purposes. In *Proceedings of JIST*.

Nazreena Rahman and Bhogeswar Borah. 2023. Query-based extractive text summarization using sense-oriented semantic relatedness measure. *Arabian Journal for Science and Engineering*.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. 2012. Semantic cosine similarity. In *Proceedings of ICAST*.

Marcus Rohrbach, Michael Stark, György Szarvas, Iryna Gurevych, and Bernt Schiele. 2010. What helps where–and why? semantic relatedness for knowledge transfer. In *Proceedings of CVPR*.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of SemEval*.

Louis L Thurstone. 1994. A law of comparative judgment. *Psychological review*.

Seid Muhie Yimam, Abinew Ali Ayele, Gopalakrishnan Venkatesh, Ibrahim Gashaw, and Chris Biemann. 2021. Introducing various semantic models for amharic: Experimentation and evaluation with multiple tasks and datasets. *Future Internet*, 13.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*.

A Appendix

Algerian Arabic (arq) - (Rank 19)		
Models	Dev SC	Test SC
TF-IDF + EN	0.43	0.33
PPMI + EN	0.44	0.34
DziriBERT + EN	0.44	0.35
LaBSE + EN	0.46	0.36
TF-IDF + LR	0.45	0.34
PPMI + LR	0.46	0.37
DziriBERT + LR	0.48	0.37
LaBSE + LR	0.48	0.38
Wt. (Dev. SC) Ensemble	0.49	0.40

Amharic (amh) - (Rank 12)		
Models	Dev SC	Test SC
TF-IDF + EN	0.67	0.74
PPMI + EN	0.68	0.76
AmRoBERTa + EN	0.68	0.76
LaBSE + EN	0.68	0.77
TF-IDF + LR	0.67	0.75
PPMI + LR	0.69	0.77
AmRoBERTa + LR	0.70	0.78
LaBSE + LR	0.70	0.78
Wt. (Dev. SC) Ensemble	0.71	0.79

English (eng) - (Rank 11)		
Models	Dev SC	Test SC
TF-IDF + EN	0.76	0.78
PPMI + EN	0.78	0.80
RoBERTa + EN	0.79	0.82
LaBSE + EN	0.80	0.82
TF-IDF + LR	0.78	0.81
PPMI + LR	0.79	0.82
RoBERTa + LR	0.80	0.83
LaBSE + LR	0.80	0.83
Wt. (Dev. SC) Ensemble	0.81	0.84

Hausa (hau) - (Rank 18)		
Models	Dev SC	Test SC
TF-IDF + EN	0.31	0.42
PPMI + EN	0.33	0.45
HauRoBERTa + EN	0.34	0.46
LaBSE + EN	0.34	0.46
TF-IDF + LR	0.32	0.41
PPMI + LR	0.33	0.45
HauRoBERTa + LR	0.35	0.46
LaBSE + LR	0.35	0.47
Wt. (Dev. SC) Ensemble	0.36	0.48

Kinyarwanda (kin) - (Rank 18)		
Models	Dev SC	Test SC
TF-IDF + EN	0.23	0.31
PPMI + EN	0.25	0.33
KinyaBERT + EN	0.25	0.34
LaBSE + EN	0.25	0.34
TF-IDF + LR	0.25	0.33
PPMI + LR	0.25	0.33
KinyaBERT + LR	0.25	0.34
LaBSE + LR	0.27	0.35
Wt. (Dev. SC) Ensemble	0.28	0.37

Marathi (mar) - (Rank 19)		
Models	Dev SC	Test SC
TF-IDF + EN	0.65	0.76
PPMI + EN	0.67	0.77
MarathiBERT + EN	0.68	0.80
LaBSE + EN	0.68	0.79
TF-IDF + LR	0.67	0.79
PPMI + LR	0.67	0.80
MarathiBERT + LR	0.69	0.81
LaBSE + LR	0.69	0.81
Wt. (Dev. SC) Ensemble	0.70	0.82

Moroccan Arabic (ary) - (Rank 21)		
Models	Dev SC	Test SC
TF-IDF + EN	0.41	0.30
PPMI + EN	0.43	0.33
DarijaBERT + EN	0.44	0.34
LaBSE + EN	0.45	0.34
TF-IDF + LR	0.44	0.34
PPMI + LR	0.45	0.35
DarijaBERT + LR	0.46	0.36
LaBSE + LR	0.46	0.36
Wt. (Dev. SC) Ensemble	0.48	0.38

Spanish (esp) - (Rank 19)		
Models	Dev SC	Test SC
TF-IDF + EN	0.58	
PPMI + EN	0.58	
SpanishBERT + EN	0.61	
LaBSE + EN	0.63	
TF-IDF + LR	0.62	
PPMI + LR	0.62	
SpanishBERT + LR	0.63	
LaBSE + LR	0.63	
Wt. (Dev. SC) Ensemble	0.66	0.65

Telugu (tel) - (Rank 13)		
Models	Dev SC	Test SC
TF-IDF + EN	0.71	0.72
PPMI + EN	0.74	0.76
TeluguBERT + EN	0.75	0.77
LaBSE + EN	0.75	0.77
TF-IDF + LR	0.74	0.75
PPMI + LR	0.74	0.76
TeluguBERT + LR	0.75	0.77
LaBSE + LR	0.76	0.78
Wt. (Dev. SC) Ensemble	0.78	0.80

Table 4: Results of Track A (Supervised) (EN : ElasticNet, LR : Linear Regression, SC : Spearman correlation)

Afrikaans (afr) - (Rank 4)			**Indonesian (ind) - (Rank 6)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF	0.01	0.73	TF-IDF	0.31	0.33
PPMI	0.02	0.73	PPMI	0.32	0.35
AfricanBERTa	0.02	0.74	IndoBERT	0.33	0.36
Ensemble	0.02	0.76	Ensemble	0.35	0.38
Algerian Arabic (arq) - (Rank 3)			**Kinyarwanda (kin) - (Rank 1)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF	0.45	0.36	TF-IDF	0.13	0.42
PPMI	0.48	0.38	PPMI	0.14	0.44
DziriBERT	0.49	0.40	KinyaBERT	0.14	0.45
Ensemble	0.52	0.42	Ensemble	0.15	0.46
Amharic (amh) - (Rank 3)			**Modern Standard Arabic (arb) - (Rank 4)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF	0.61	0.61	TF-IDF	0.40	0.37
PPMI	0.63	0.63	PPMI	0.41	0.38
AmRoBERTa	0.66	0.65	ArabicBERT	0.41	0.39
Ensemble	0.67	0.66	Ensemble	0.42	0.40
English (eng) - (Rank 8)			**Moroccan Arabic (ary) - (Rank 2)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF	0.63	0.72	TF-IDF	0.61	0.51
PPMI	0.65	0.74	PPMI	0.63	0.54
RoBERTa	0.66	0.75	DarijaBERT	0.63	0.55
Ensemble	0.68	0.77	Ensemble	0.65	0.56
Hausa (hau) - (Rank 2)			**Punjabi (pan) - (Rank 2)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF	0.42	0.45	TF-IDF	0.03	0.01
PPMI	0.45	0.47	PPMI	0.03	0.01
HauRoBERTa	0.46	0.48	PunjabiBERT	0.04	0.02
Ensemble	0.47	0.50	Ensemble	0.04	0.02
Hindi (hin) - (Rank 7)			**Spanish (esp) - (Rank 4)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF	0.58	0.53	TF-IDF	0.57	
PPMI	0.58	0.54	PPMI	0.58	
HindiBERT	0.60	0.56	SpanishBERT	0.59	
Ensemble	0.61	0.57	Ensemble	0.60	0.66

Table 5: Results for Track B (Unsupervised) (EN : ElasticNet, LR : Linear Regression, SC : Spearman Correlation)

Afrikaans (afr) - (Rank 11)			**Indonesian (ind) - (Rank 11)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF + EN	0.07	0.33	TF-IDF + EN	0.24	0.10
PPMI + EN	0.08	0.35	PPMI + EN	0.25	0.11
ArabicBERT + EN	0.09	0.35	RoBERTa-tagalog + EN	0.27	0.12
TF-IDF + LR	0.08	0.34	TF-IDF + LR	0.26	0.11
PPMI + LR	0.10	0.36	PPMI + LR	0.27	0.12
ArabicBERT + LR	0.10	0.37	RoBERTa-tagalog + LR	0.27	0.13
Ensemble	0.11	0.38	Ensemble	0.29	0.13
Algerian Arabic (arq) - (Rank 9)			**Kinyarwanda (kin) - (Rank 12)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF + EN	0.25	0.17	TF-IDF + EN	0.22	0.03
PPMI + EN	0.27	0.19	PPMI + EN	0.23	0.04
AfricanBERTa + EN	0.27	0.20	ArabicBERT + EN	0.26	0.06
TF-IDF + LR	0.27	0.19	TF-IDF + LR	0.24	0.05
PPMI + LR	0.28	0.21	PPMI + LR	0.25	0.06
AfricanBERTa + LR	0.29	0.21	ArabicBERT + LR	0.26	0.07
Ensemble	0.30	0.22	Ensemble	0.28	0.08
Amharic (amh) - (Rank 9)			**Modern Standard Arabic (arb) - (Rank 8)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF + EN	0.06	0.08	TF-IDF + EN	0.21	0.15
PPMI + EN	0.09	0.09	PPMI + EN	0.24	0.18
ArabicBERT + EN	0.09	0.10	AfricanBERTa + EN	0.25	0.18
TF-IDF + LR	0.09	0.10	TF-IDF + LR	0.22	0.16
PPMI + LR	0.10	0.11	PPMI + LR	0.25	0.18
ArabicBERT + LR	0.10	0.12	AfricanBERTa + LR	0.26	0.19
Ensemble	0.11	0.13	Ensemble	0.27	0.21
English (eng) - (Rank 9)			**Moroccan Arabic (ary) - (Rank 10)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF + EN	0.25	0.26	TF-IDF + EN	0.09	0.14
PPMI + EN	0.26	0.27	PPMI + EN	0.12	0.17
SpanishBERT + EN	0.28	0.29	AfricanBERTa + EN	0.12	0.18
TF-IDF + LR	0.26	0.28	TF-IDF + LR	0.10	0.15
PPMI + LR	0.27	0.28	PPMI + LR	0.13	0.17
SpanishBERT + LR	0.28	0.30	AfricanBERTa + LR	0.14	0.19
Ensemble	0.29	0.31	Ensemble	0.15	0.20
Hausa (hau) - (Rank 12)			**Punjabi (pan) - (Rank 5)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF + EN	0.08	0.06	TF-IDF + EN	0.01	0.01
PPMI + EN	0.09	0.07	PPMI + EN	0.02	0.01
ArabicBERT + EN	0.11	0.08	HindiBERT + EN	0.03	0.02
TF-IDF + LR	0.09	0.07	TF-IDF + LR	0.02	0.01
PPMI + LR	0.10	0.07	PPMI + LR	0.03	0.02
ArabicBERT + LR	0.11	0.09	HindiBERT + LR	0.04	0.02
Ensemble	0.12	0.10	Ensemble	0.04	0.02
Hindi (hin) - (Rank 9)			**Spanish (esp) - (Rank 10)**		
Models	Dev SC	Test SC	Models	Dev SC	Test SC
TF-IDF + EN	0.48	0.43	TF-IDF + EN	0.39	
PPMI + EN	0.51	0.47	PPMI + EN	0.40	
BanglaBERT + EN	0.53	0.49	roBERTa + EN	0.43	
TF-IDF + LR	0.50	0.46	TF-IDF + LR	0.41	
PPMI + LR	0.52	0.48	PPMI + LR	0.42	
BanglaBERT + LR	0.53	0.50	roBERTa + LR	0.44	
Ensemble	0.55	0.51	Ensemble	0.45	0.56

Table 6: Results for Track C (Cross-lingual) (EN : ElasticNet, LR : Linear Regression, SC : Spearman Correlation)

RiddleMasters at SemEval-2024 Task 9: Comparing Instruction Fine-tuning with Zero-Shot Approaches

Kejsi Take
New York University
Brooklyn, NY
kejsitake@nyu.edu

Chau Tran
New York University
Brooklyn, NY
chau.tran@nyu.edu

Abstract

This paper describes our contribution to SemEval 2023 Task 9: Brainteaser. We compared multiple zero-shot approaches using GPT-4, the state of the art model with Mistral-7B, a much smaller open-source LLM. While GPT-4 remains a clear winner in all the zero-shot approaches, we show that fine-tuning Mistral-7B can achieve comparable, even though marginally lower results.

1 Introduction

Traditionally, the natural language processing (NLP) community has focused on tasks that require objective and complex reasoning. On the other hand, puzzles that defy traditional ways of reasoning have been less explored. Brainteaser, a task at SemEval 2024 (Jiang et al., 2024), aims to fill this gap by investigating the abilities of large language models (LLMs) in more abstract and creative thinking. This competition consists of two sub-tasks: sentence puzzle (SP) and word puzzle (WP). According to the task description , sentence puzzles are brainteasers where the entire sentence snippet defies common sense. Similarly, word puzzles are puzzles where the answer violates the default meaning of the word and focus on the letter composition of the question.

In this work, we investigate a set of zero-shot approaches and compare them with a fine-tuned version of Mistral-7B, an open source 7 billion LLM (Jiang et al., 2023a). For the zero-shot approaches, we compare Mistral-7B with GPT-4, the state of the art transformer model, across various prompts. We find that one-shot approaches using GPT-4 produces the best results across both our tasks. However, tweaking the prompts results in significant accuracy increases for Mistral-7B. We also find that fine-tuned Mistral-7B is the second best model in the sentence puzzle sub-task, indicating that instruction fine-tuning may be a way to get better results with smaller models.

2 Background

The NLP task most related to this competition is question answering (QA), as all riddles consists of a question and multiple potential answers. Question answering has been the focus of extensive prior work (Soares and Parreiras, 2020). Typically, question answering systems consist of three main components: (1) *question processing*, (2) *document processing*, and (3) *answer processing* (Bhoir and Potey, 2014; Soares and Parreiras, 2020). The main goal of the question processing is to extract the keywords from the query so they can be parsed to the document processing component, as well as to identify the type of answer that we need to return (Parsing, 2009). The goal for the document processing system is information retrieval (IR), based on the keywords collected from the previous component. Typically, the IR system's job is to identify a subset of documents relevant to the keywords identified previously (Malik et al., 2013; Gupta and Gupta, 2012).

As the desired output needs to be accurate and succinct, the IR system needs to further break down the relevant documents into smaller units such as passages, paragraphs, or sentences. The final stage of question answering is answer processing, which involves formulating the desired answer based on the knowledge previously retrieved, using a process called span labeling (Parsing, 2009): given a passage, identifying the span of text that can be used to answer the question. These components are largely suitable for answering questions in a way that utilizes straightforward information processing and logical thinking, but struggle against question answering tasks that require creative responses or common-sense reasoning, such as solving puzzles and brainteasers (Jiang et al., 2021, 2023b).

With the recent breakthroughs of LLMs such as BERT (Devlin et al., 2019) or GPT-3 (Brown et al., 2020), we have seen exceptional capabili-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1391–1396

ties of these language models in solving QA tasks, as well as their exhibition of complex reasoning abilities (Hu et al., 2024; Creswell et al., 2022). However, when it comes to creative thinking and common sense reasoning, large language models achieve limited results (Ding et al., 2023; Zhou et al., 2020; AlKhamissi et al., 2022). As such, (Jiang et al., 2023b) generate a dataset of brain-teasers to benchmark the performance pf state-of-the-art LLMs in answering puzzles and brain-teasers, as a way to test their lateral thinking capabilities. Our work aims to contribute to this domain, by evaluating the performance of multiple zero-shot approaches and our version of fine-tuned Mistral-7B language model (Jiang et al., 2023a) on the same dataset (Jiang et al., 2023b).

Dataset. The authors generated the initial Brain-teaser dataset by crawling the puzzles from the internet. However, recent work has shown that memorization is a common problem with LLMs (Carlini et al., 2022). To evaluate lateral thinking instead of memorization, the authors used two reconstruction strategies (semantic and context) to create variants of each puzzle. Semantic reconstruction rephrases the original question and was created via an open-source rephrasing tool (Jiang et al., 2023b). In contrast, context reconstruction was achieved through a combination of GPT-4 prompts and human annotators (Jiang et al., 2023b).

3 System Overview

In this section, we first describe the train and test datasets. Later, we describe our proposed approaches, detailing the prompts used in the zero-shot approaches and the fine-tuning methodology.

3.1 Dataset Description

As mentioned above, we used the provided dataset (Jiang et al., 2023b) which consists of 1,119 data samples, including its reconstruction variants. The questions were divided into two sub-tasks, Sentence Puzzle and Word Puzzle, and further distributed into more than 80 different areas/topics. For more details about the dataset distribution, please refer to (Jiang et al., 2023b).

Train-test split. The data provided by the organizers was split 80:20 between the train and test set. In general there are more examples of sentence puzzles (627 total) than word puzzles (492 total).

Label Distribution. To investigate model bias, we investigated the distribution of labels. As shown in Figure 1, the correct answers are not evenly distributed among all options. In fact, the 4^{th} label (i.e., "None of the above") is the minority label. This label is particularly rare in the train set for word puzzles (9/397) and does not occur in the test set for the same task.

	Sentence Puzzle		Word Puzzle	
train	507	80.8%	396	80.4%
test	120	19.1%	96	19.5%
total	627		492	

Table 1: Number of samples in test and train data.

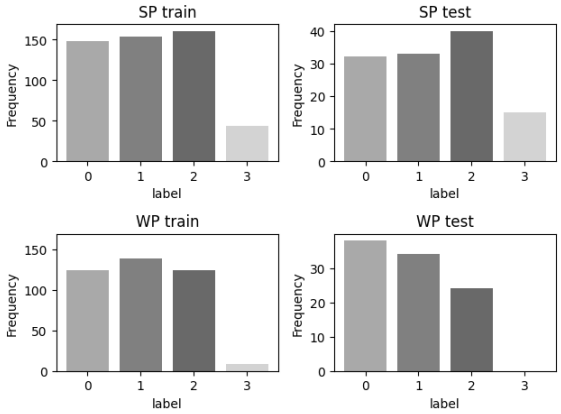


Figure 1: Distribution of the labels between answer choices (answer choices are encoded as 1, 2, 3, 4)

3.2 Zero-Shot Prompting

Given that prior work found that common sense models are not more effective than zero-shot approaches (Jiang et al., 2023b), in this paper we focused only on the latter. We started our evaluation by experimenting with zero-shot solutions. We wanted to compare Mistral-7B with GPT-4, the state of the art transformer model. To provide a thorough evaluation, we experimented with three prompting strategies, which we include in Appendix A. The first one is identical to the one provided by the competition organizers. To formulate the second prompt we leveraged the fact that all riddles contain a "None of the above." answer. This way, the second prompt provides three answer options (excluding "None of the above."). If none of the three answers is returned, we consider that to be "None of the above." In the third prompt, we tried to guide the model to consider all answer options, emulating an approach similar to zero-shot Chain-of-thought (CoT) prompting (Kojima et al., 2022). Similar to the second prompt, here we limit

			Instance-based			Group-based		
Category		**Model**	**Original**	**Semantic**	**Context**	**Ori & Sem**	**All**	**Overall**
Sentence Puzzle								
Random	-	-	0.175	0.150	0.175	0.000	0.000	0.167
	P1	**GPT-4**	0.825	0.700	0.725	0.675	0.600	0.750
		Mistral	0.275	0.250	0.225	0.225	0.125	0.250
Zero-Shot	**P2**	**GPT-4**	0.850	**0.800**	0.700	0.800	0.625	0.783
		Mistral	0.500	0.500	0.350	0.425	0.250	0.450
	P3*	**GPT-4**	**0.925**	0.750	0.775	**0.750**	**0.675**	**0.817**
Finetuning		**Mistral**	0.800	0.775	**0.800**	0.725	0.650	0.792
Word Puzzle								
Random	-	-	0.094	0.250	0.219	0.031	0.031	0.188
	P1	**GPT-4**	0.625	0.531	0.656	0.438	0.312	0.604
		Mistral	0.031	0.062	0.094	0.031	0.031	0.062
Zero-Shot	**P2**	**GPT-4**	**0.906**	**0.906**	**0.906**	**0.875**	**0.781**	**0.906**
		Mistral	0.594	0.656	0.625	0.469	0.312	0.625
	P3*	**GPT-4**	0.875	0.906	0.812	0.844	0.719	0.865
Finetuning		**Mistral**	0.844	0.844	0.844	0.781	0.656	0.844

Table 2: Results for the two BRAINTEASER subtasks across all models, prompts (P1, P2, P3) and metrics. Ori = Original, Sem = Semantic, All = Original + Semantic + Context. The best performance among all models is in bold. The random is answer assigned by random choice where the four options have equal probability to be selected. For prompt 3, Mistral-7B did not generate any meaningful responses and therefore we do not include it in this evaluation.

the choices to the three options. Further, we prompt the model to respond "None" if none of the three options is not the answer.

3.3 Instruction-based Fine-tuning

We fine-tuned a sharded version of Mistral-7B[1]. Mistral-7B is an LLM with 7.3 billion parameters. Mistral-7B uses grouped-query attention for faster inference and sliding-window attention to handle longer sequence (Jiang et al., 2023a). We used instruction based fine-tuning, a type of fine-tuning where instructions are used to define downstream tasks. In our case, the instruction was formed by the question and the sample answers.

We fine-tuned the model using Google Colab and used Low-Rank Adaptation (LoRA) (Hu et al., 2021) to make fine-tuning more efficient. LoRA freezes the pre-trained model weights and using rank decomposition matrices into each layer to reduce the number of trainable parameters.

Training Parameters. When fine-tuning with LoRA, one of the parameters is a list of specific layers in the model architecture that will undergo decomposition. While limiting only to attention layers may reduce training time, we targeted all linear layers, as prior work (Dettmers et al., 2024) suggests that this might provide better results. The other significant LoRA parameter is r, the rank of matrices updated during adaptation. However, it has been shown that the value of r does not improve adaptation quality between a certain point[2] and therefore we keep $r = 8$. These approaches result in 21M trainable parameters (0.29%) instead of a total of 7B.

4 Results

The results for all the experiments are included in Table 2. In this section, we discuss and compare all the approaches.

4.1 Zero-Shot Prompting

In the zero-shot experiments, Mistral-7B generally performs worse than GPT-4. This is not surprising, as it is a much smaller model (7 billion parameters vs 1.76 trillion). Further, zero-shot approaches with prompt 2 and prompt 3 perform better than the one with prompt 1. These approaches are also the improvement from the zero-shot approaches described in the paper introducing the Brainteaser dataset (Jiang et al., 2023b).

[1]https://huggingface.co/alexsherstinsky/Mistral-7B-v0.1-sharded

[2]https://www.databricks.com/blog/efficient-fine-tuning-lora-guide-llms

	Sentence Puzzle					Word Puzzle				
Model	**Avg. F1**	**F1(1)**	**F1(2)**	**F1(3)**	**F1(4)**	**Avg. F1**	**F1(1)**	**F1(2)**	**F1(3)**	**F1(4)**
GPT-4(P2)	0.783	0.817	0.833	0.810	0.334	**0.906**	**0.949**	**0.912**	0.844	0.000
GPT-4(P3)	**0.817**	**0.881**	**0.879**	**0.849**	0.571	0.865	0.914	0.889	**0.939**	0.000
FT-Mistral	0.792	0.779	0.831	0.805	**0.706**	0.844	0.853	0.862	0.808	0.000

Table 3: Average F1-scores overall and for all answer choices (1,2,3,4) for the three best performing models. It is visible that the zero-shot approaches on GPT-4, the best performing from Table 2, are biased towards the first three answers, resulting on a lower F1-score for the 4th answer (None of the above.)

Sentence Puzzles. We find that tweaking the prompt results in performance improvements. Using prompt 2 instead of prompt 1 results in a marginal increase (3%) of the overall performance and using prompt 3 results in about 6% overall improvement. In this case, the improvement is more significant in the original brainteasers with about 10%.

Word Puzzles Prompt choice seems to have a more significant impact in word puzzles. As visible in Figure 2, using prompt 2 and prompt 3 instead of prompt 1, results in respectively 26% and 30% overall accuracy increase.

4.2 Instruction-based Fine-tuning

According to Table 2 the fine-tuned model is only marginally worse that the zero-shot approaches (prompt 2 & 3) in the sentence puzzles sub-task. However, in the word puzzles sub-task, it performs 6% worse overall than the best performing zero-shot approach. In summary, the three best performing models are GPT-4 zero-shot approaches (prompt 2 & 3) and the instruction fine-tuned model.

However, due to the imbalanced distribution of correct answers between different labels, we also looked into the F1 scores of different labels. We calculated F1 scores overall and for each label and includes the results of this comparison for the three best performing models in Table 3. The table indicates that the zero-shot approaches result in lower scores in the 4th label ("None of the above"). Indeed, the F1 score for this label is improved in the fine-tuned approach. In summary, this finding highlights the need to investigate solutions and metrics beyond the simple accuracy metrics.

5 Limitations and Future Work

While we tried to explore various prompts for our zero-shot approaches, there is a possibility that further experiments might reveal more effective techniques. Future work could explore additional prompts as well as look into automating the prompt-search process. Another area of potential improvements would be the exploration of additional datasets, especially those that include similar riddles based on lateral thinking. Lastly, future work could explore additional fine-tuning techniques and discover if the accuracy can be further improved.

6 Conclusions

We present a comparison of zero-shot approaches with instruction fine-tuning within SemEval-2024 Task 9. Our experiments applied a variety of best practices for prompt engineering to explore the potential of zero-shot approaches in tasks that require lateral thinking and reasoning. We find that upon iterating over multiple prompts, zero-shots approaches using GPT-4 remain the solution that results in higher accuracy. However, instruction fine-tuned Mistral-7B provides a second best alternative in the sentence puzzle sub-task.

References

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*.

Varsha Bhoir and MA Potey. 2014. Question answering system: A heuristic approach. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 165–170. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang.

2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zijian Ding, Arvind Srinivasan, Stephen MacNeil, and Joel Chan. 2023. Fluid transformers and creative analogies: Exploring large language models' capacity for augmenting cross-domain analogical creativity. In *Proceedings of the 15th Conference on Creativity and Cognition*, pages 489–505.

Poonam Gupta and Vishal Gupta. 2012. A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4).

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Sheng Bi, Tongtong Wu, and Jeff Z Pan. 2024. Benchmarking large language models in complex question answering attribution using knowledge graphs. *arXiv preprint arXiv:2401.14640*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023b. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Nidhi Malik, Aditi Sharan, and Payal Biswas. 2013. Domain knowledge enriched framework for restricted domain question answering system. In *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–7.

Constituency Parsing. 2009. Speech and language processing. *Power Point Slides*.

Marco Antonio Calijorne Soares and Fernando Silva Parreiras. 2020. A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, 32(6):635–646.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9733–9740.

A Appendix A

A.1 Prompt 1

```
Please pick the best choice for the
brain teaser. Each brain teaser has
only one possible solution including
the choice none of above, answer
should only provide the choice:
Question: {}
Choice:
(A) {}
(B) {}
(C) {}
(D) {}
Answer:
```

A.2 Prompt 2

```
Below is an instruction that describes
a riddle, paired with four choices.
Choose the option that appropriately
answers the riddle.
### Riddle:
{}
### Options:
1 - {}
2 - {}
3 - {}
### Instruction:
In the end, print the number of the
correct answer between these tags:
<answer> </answer>:
```

A.3 Prompt 3

```
You are a great riddlemaster that is
very helpful in solving riddles.
Solve the following riddle:
{}
Consider each of the following
answers and provide reasons why
they are or are not correct.
If none is correct, print "None".
1) {}
2) {}
3) {}
In the end print the correct
answer between these tags:
<answer> </answer>
```

IITK at SemEval-2024 Task 2: Exploring the Capabilities of LLMs for Safe Biomedical Natural Language Inference for Clinical Trials

Shreyasi Mandal **Ashutosh Modi**
Indian Institute of Technology, Kanpur (IIT Kanpur)
{shreyansi, ashutoshm}@cse.iitk.ac.in

Abstract

Large Language models (LLMs) have demonstrated state-of-the-art performance in various natural language processing (NLP) tasks across multiple domains, yet they are prone to shortcut learning and factual inconsistencies. This research investigates LLMs' robustness, consistency, and faithful reasoning when performing Natural Language Inference (NLI) on breast cancer Clinical Trial Reports (CTRs) in the context of SemEval 2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. We examine the reasoning capabilities of LLMs and their adeptness at logical problem-solving. A comparative analysis is conducted on pre-trained language models (PLMs), GPT-3.5, and Gemini Pro under zero-shot settings using Retrieval-Augmented Generation (RAG) framework, integrating various reasoning chains. The evaluation yields an F1 score of **0.69**, consistency of **0.71**, and a faithfulness score of **0.90** on the test dataset.

1 Introduction

Clinical trials serve as essential endeavors to evaluate the effectiveness and safety of new medical treatments, playing a pivotal role in advancing experimental medicine. Clinical Trial Reports (CTRs) detail the methodologies and outcomes of these trials, serving as vital resources for healthcare professionals in designing and prescribing treatments. However, the sheer volume of CTRs (e.g., exceeding 400,000 and proliferating) presents a challenge for comprehensive literature assessment when developing treatments (Bastian et al., 2010). Natural Language Inference (NLI) (Bowman et al., 2015) emerges as a promising avenue for large-scale interpretation and retrieval of medical evidence bridging recent findings to facilitate personalized care (DeYoung et al., 2020; Sutton et al., 2020). The SemEval 2024 Task 2 on the Natural Language Inference for Clinical Trials (NLI4CT) (Jullien et al., 2024) revolves around annotating statements extracted from breast cancer CTRs[1] and determining the inference relation between these statements and corresponding sections of the CTRs, such as Eligibility criteria, Intervention, Results, and Adverse events. By systematically intervening in the statements, targeting numerical, vocabulary, syntax, and semantic reasoning, the task aims to investigate Large Language Models (LLM)s' consistency and faithful reasoning capabilities.

In this paper, we experiment with Gemini Pro (Team et al., 2023), GPT-3.5 (Brown et al., 2020), Flan-T5 (Longpre et al., 2023) and several pre-trained language models (PLMs) trained on biomedical datasets, namely BioLinkBERT (Yasunaga et al., 2022), SciBERT (Beltagy et al., 2019), ClinicalBERT (Huang et al., 2019). We conducted zero-shot evaluations of Gemini Pro and GPT-3.5, employing Retrieval Augmented Generation (RAG) framework (Lewis et al., 2020) integrating Tree of Thoughts (ToT) reasoning (Yao et al., 2023) facilitating multiple reasoning paths. Our experiments involved applying various instruction templates to guide the generation process. These templates were refined through manual comparison of the labels within the training dataset against those generated by the models. The PLMs were fine-tuned on the provided training dataset, while the Flan-T5 model was assessed under zero-shot conditions.

Gemini Pro emerged as the top-performing model among all the experimented models, achieving an F1 score of **0.69**, with consistency and faithfulness scores of **0.71** and **0.90**, respectively, on the official test dataset. Notably, a comparative analysis between GPT-3.5 and Gemini Pro revealed shortcomings in GPT-3.5's performance, particularly in instances requiring numerical reasoning. For detailed examination of such instances, please

[1] https://clinicaltrials.gov/ct2/home

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1397–1404

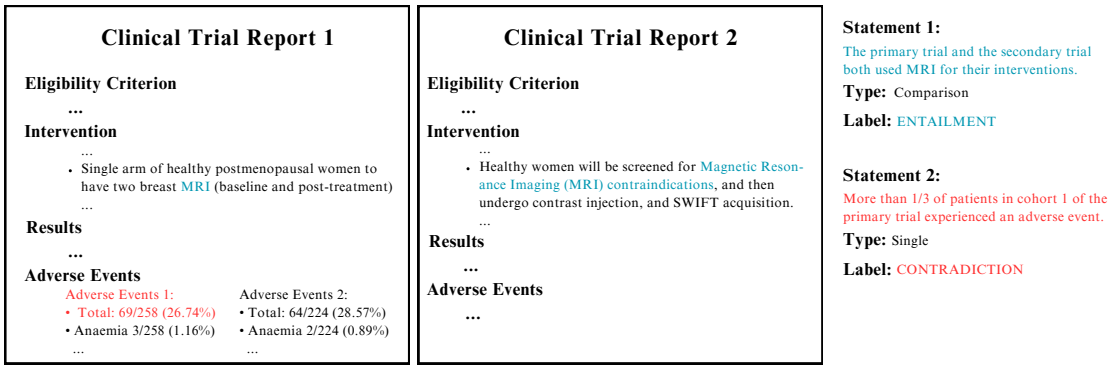


Figure 1: Examples of the dataset used in the NLI4CT task. Statement 1 compares the *Intervention* section from two different clinical trial reports, while statement 2 is based on the *Adverse Events* section of the first clinical trial report. The evaluation of the first statement requires textual inference skills, while the second requires numerical inference skills.

refer to Appendix A, where an example showcases GPT-3.5's accurate inference yet inadequate conclusion. The code to reproduce the experiments mentioned in this paper is publicly available.[2]

2 Background

2.1 Related Work

Pretrained Language Models (PLMs) and Large Language Models (LLMs) exhibit the potential to yield promising outcomes in the biomedical domain due to their ability to comprehend and process complex medical data effectively. BioLinkBERT (Yasunaga et al., 2022), pre-trained on PubMed[3], utilizes hyperlinks within documents. It has attained state-of-the-art (SOTA) performance across a wide range of tasks and various medical NLP benchmarks, namely BLURB (Gu et al., 2021) and BioASQ (Nentidis et al., 2020). SciBERT (Beltagy et al., 2019) is trained on scientific publications from the biomedical domain in Semantic Scholar[4]. ClinicalBERT (Huang et al., 2019) is trained using clinical text data sourced from approximately 2 million clinical notes contained within the MIMIC-III database (Johnson et al., 2016). Kanakarajan and Sankarasubbu (2023) employed a fine-tuned Flan-T5-xxl model with instruction tuning, achieving an F1 score of 0.834 on the SemEval 2023 Task 7 (Jullien et al., 2023a,b). Zhou et al. (2023) performed joint semantics encoding of the clinical statements followed by multi-granularity inference through sentence-level and token-level encoding, getting an F1 score of 0.856. Although these models have achieved high performance, there remains a need for further investigation into their application in vital areas such as real-world clinical trials.

GPT-3.5, developed by OpenAI[5] and comprising 175 billion parameters, uses alternating dense and locally banded sparse attention patterns in the transformer layers (Child et al., 2019; Wolf et al., 2020). The token size limit for GPT-3.5 (free tier) is 4,096. **Gemini Pro**, developed by Google DeepMind[6] uses decoder-only transformers (Vaswani et al., 2017) and multi-query attention (Shazeer, 2019) with a context window length of 32,768 tokens.

Data	Number of Samples	Labels	
		Entailment	Contradiction
train	1700	850	850
dev	200	100	100
test	5500	1841	3659

Table 1: The number of samples in each subset of the data. The distribution of the labels between the train and the development set is even. Note: The test set labels were made public after the completion of the task.

2.2 Task and Dataset Description

The NLI4CT task (Jullien et al., 2024) focuses on textual entailment based on a collection of breast cancer CTRs, statements, explanations and labels annotated by domain expert annotators. The CTRs are in English. The CTRs are segmented into four

[2]https://github.com/Exploration-Lab/IITK-SemEval-2024-Task-2-Clinical-NLI
[3]https://pubmed.ncbi.nlm.nih.gov
[4]https://www.semanticscholar.org
[5]https://openai.com
[6]https://deepmind.google

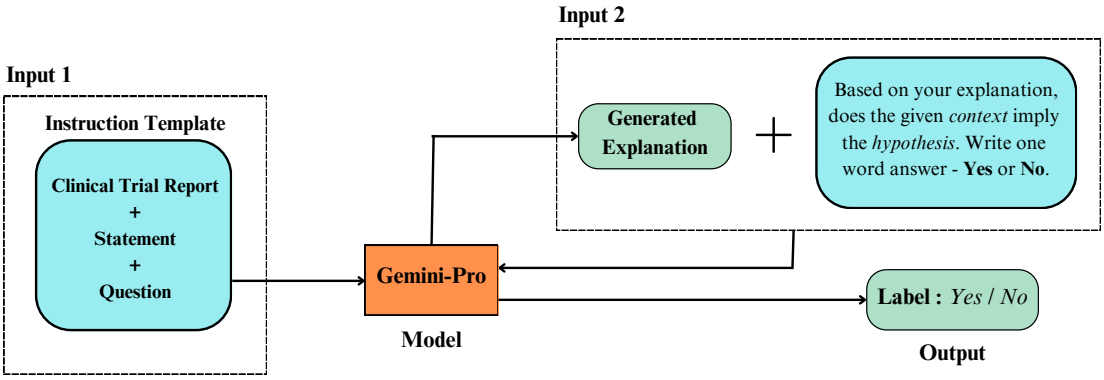


Figure 2: An overview of the proposed system architecture used for the NLI4CT Task

sections - eligibility criteria, intervention details, results, and adverse events. The statements, with an average length of 19.5 tokens, make claims about the information contained in one of the sections of a CTR or compare the same section from two different CTRs as seen in Figure 1. The task involves determining the inference relation (*entailment* or *contradiction*) between CTR-statement pairs. The dataset consists of 999 Clinical Trial Reports (CTRs) and 7400 annotated statements, which are divided into train, development and test sets. Table 1 provides statistics for the dataset.

3 System Overview

LLMs such as GPT-3 (Brown et al., 2020) and Gemini Pro (Team et al., 2023) have shown remarkable performances across various tasks. For the NLI4CT task, we have experimented with Gemini Pro, GPT-3.5, Flan-T5 (Longpre et al., 2023), BioLinkBERT (Yasunaga et al., 2022), SciBERT (Beltagy et al., 2019), ClinicalBERT (Huang et al., 2019) and ClinicalTrialBioBert-NLI4CT[7]. The performance of the different models is shown in Figure 7. Zero-shot evaluation was done on Gemini Pro and GPT-3.5, Flan-T5 was instruction fine-tuned following Kanakarajan and Sankarasubbu (2023), and the rest of the models were trained on the given train and development dataset. Gemini Pro and GPT-3.5 were considered for further experimentation because of their superior performance.

The proposed system utilizes structured instruction templates and multi-turn conversation techniques to generate explanations and labels for the statements provided as input, as shown in Figure 2.

Reasoning is an essential ability required by an LLM to solve complex problems (Qiao et al., 2022). Tree of Thoughts (ToT) framework (Yao et al., 2023) and Chain-of-Thought (CoT) reasoning (Wei et al., 2022) is integrated into the models, facilitating multiple reasoning paths.

3.1 Reasoning Frameworks

Chain-of-Thought (CoT) prompting (Wei et al., 2022) has demonstrated promising results in improving the reasoning abilities of LLMs. To evaluate Gemini Pro and GPT-3.5, we used Zero-shot-CoT (Kojima et al., 2022) prompt reasoning without requiring few-shot demonstrations. The phrase *"Let's think step by step"* is added after the instruction as shown in Figure 3.

```
"{premise}

{statement}
Is the given statement correct based on the information
provided above? Let's think step by step."

>> {explanation by model}

"From your answer above, is the statement true or false?
Write one word answer."

>> {True / False}
```

Figure 3: Instruction template for CoT prompting

Tree-of-Thought (ToT) framework (Yao et al., 2023; Long, 2023) relies on trial and error method to solve complex reasoning tasks. It facilitates multi-round conversations and backtracking. Our system allows for three reasoning paths using the prompt shown in Figure 4.[8]

For the evaluation of the model, the input to Gemini Pro and GPT-3.5 is constructed using an instruction template containing the appropriate prompt for ToT or CoT reasoning, data from the

[7] https://huggingface.co/domenicrosati/ClinicalTrialBioBert-NLI4CT

[8] https://github.com/dave1010/tree-of-thought-prompting

Imagine three different clinical experts are answering the question given below.
All experts will write down first step of their thinking, then share it with the group.
Then all experts will go on to the next step of their thinking.
If any expert realises they're wrong at any point then they leave.
They will continue till a definite conclusion is reached.

Figure 4: Prompt for Tree of Thought reasoning

CTR which constitutes the premise and the statement or the hypothesis as shown in Figure 2. A series of two questions is presented to the model to generate both the explanation and the corresponding label. Multi-turn conversation (Zhang et al., 2018) is used to include the generated explanation as context for generating the final label. The explanation is also retained for further experimentations. The generated final label is converted as follows: *{"Yes": "Entailment", "No": "Contradiction"}*. A comparison of the performance of GPT-3.5 and Gemini Pro after integrating CoT and ToT reasoning frameworks is shown in Figure 6.

4 Experimental setup

4.1 Data Preprocessing

As discussed in Section 2.2, the statements can make claims about the information contained in one of the sections of a CTR, which is then called a "Single" statement or compare the same section from two different CTRs, called a "Comparison" statement. In "Single" statements, the term "primary" is employed to assert a claim. Evidence from the CTR is compiled into a unified text structure, exemplified as follows: *"For the primary trial participants, {primary evidences}"*. In contrast, for "Comparison" statements, the term "secondary" accompanies "primary". The evidences are then compiled as: *"For the primary trial participants, {primary evidences}. For the secondary trial participants, {secondary evidences}"*.

4.2 Hyperparameter Tuning

For Gemini Pro, the temperature of the model is set to 0.7 and the safety settings are set to "BLOCK_NONE". For GPT-3.5, the models *"gpt-3.5-turbo-0613"* and *"gpt-3.5-turbo-1106"* are used for experimentation among which *"gpt-3.5-turbo-0613"* performs considerably better. The temperature of the model is set to 0.6.

4.3 Prompt Engineering

The system was experimented with several prompts to improve its performance. The explanations generated by the model were examined manually to identify instances where the solution deviated from the correct path. The prompt *"You are a clinical expert and can seamlessly perform natural language inference"* was introduced to give the model an identity. Additionally, rules were enforced to confine the model's output within the provided context and to prevent hallucinations, achieved through the prompt: *"Please align with the context given and do not make any false assumptions of your own."* Furthermore, to integrate CoT reasoning within the ToT framework, the prompt *"Provide a step-by-step explanation of your thought process"* was introduced. The final instruction template is shown in Figure 5.

Several experiments were conducted to assess the model's performance on extracting the labels *"Entailment"* or *"Contradiction"* in the second question of the multi-turn conversation. The F1 scores for various prompts on the development set are presented in Table 2. Ultimately, Prompt 4 demonstrated the best performance and was chosen for the final pipeline.

Prompt	**F1 score**
Based on the comprehensive evaluation of the model's responses, is the given hypothesis deemed to be true or false? Write one word answer.	0.689
Does this imply that the given hypothesis is supporting the report or not? Give one word answer (Yes / No).	0.667
From your answer above, is the statement true or false? Write one word answer.	0.656
Based on your explanation, does the given context imply the hypothesis. Write one word answer.	**0.723**

Table 2: Performance of the model on the *dev* data for different prompts for extracting the labels

4.4 Evaluation Metrics

The NLI4CT task (Jullien et al., 2024) is evaluated on the basis of three metrics - F1 score, consistency and faithfulness. Faithfulness measures the accuracy of the system's predictions by evaluating its ability to predict outcomes for altered inputs correctly. If the model correctly adjusts its predictions

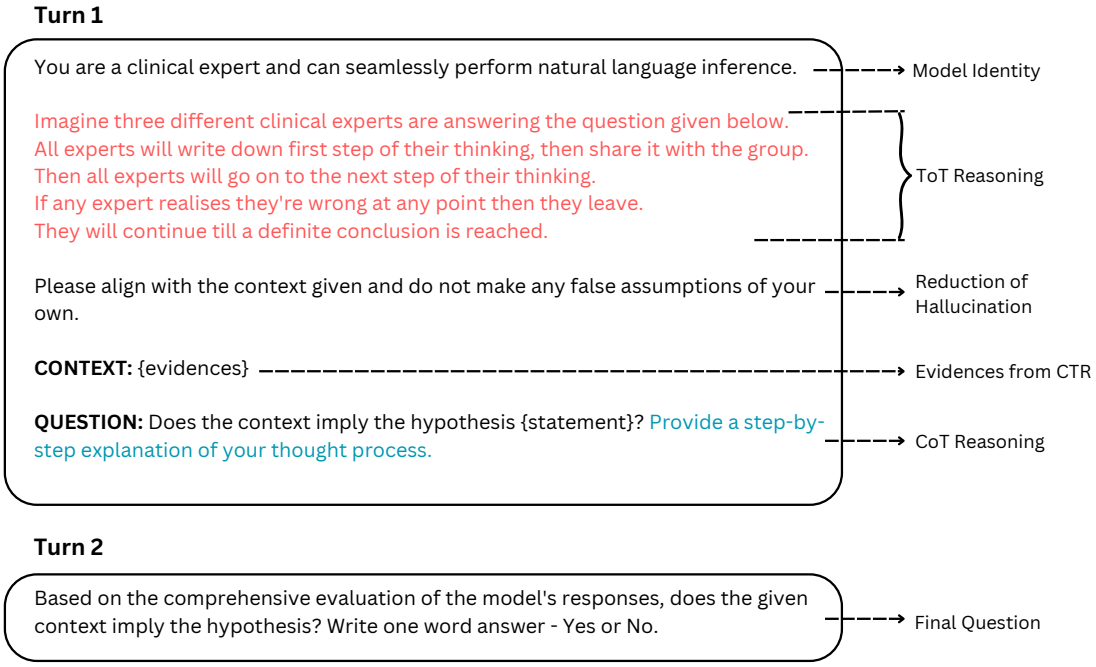


Figure 5: Final Instruction Template

in response to semantic alterations, it demonstrates higher faithfulness. On the other hand, consistency evaluates the model's ability to provide consistent predictions for semantically equivalent inputs.

5 Results

The zero-shot evaluation of Gemini Pro yielded an F1 score of **0.69**, with a consistency of **0.71** and a faithfulness score of **0.90** on the official test dataset. Our system achieved a fifth-place ranking based on the faithfulness score, a sixteenth-place ranking based on the consistency score, and a twenty-first-place ranking based on the F1 score. Gemini Pro outperforms GPT-3.5 with an improvement in F1 score by +1.9%, while maintaining almost similar consistency score. Additionally, the faithfulness score of Gemini Pro improves by +3.5% compared to GPT-3.5, as illustrated in Table 3.

Model	Base F1	Consistency	Faithfulness
Gemini Pro	**0.691**	0.712	**0.901**
GPT-3.5	0.672	**0.713**	0.866

Table 3: Results on the *test* data using Gemini Pro and GPT-3.5

The system utilizing Gemini Pro attained an F1 score of 0.72, while GPT-3.5 achieved an F1 score of 0.68 on the training dataset. Manual examination of the model-generated explanations and a comparison of the generated labels with the original labels was conducted to refine the prompts and enhance the model's responses.

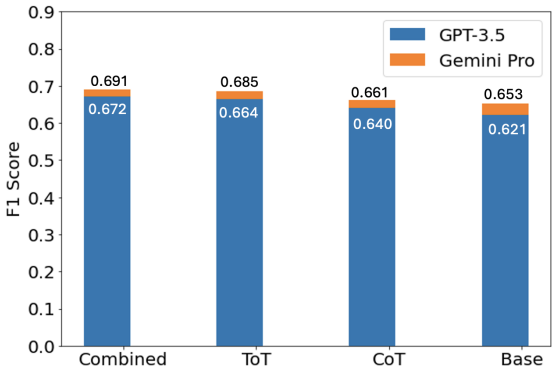


Figure 6: Comparison of the performance of Gemini Pro and GPT-3.5 without the integration of any reasoning framework, with CoT reasoning, with ToT reasoning and with both the reasoning frameworks combined.

As depicted in Figure 6, the integration of CoT reasoning led to an increase in performance for Gemini Pro and GPT-3.5 by 0.8% and 1.9%, respectively. Furthermore, upon integrating the ToT reasoning framework, the performance improved by 3.2% and 4.3%, respectively. When both ToT and CoT reasoning were integrated, the models showed an increase in performance by **3.8%** and **5.1%**, respectively, compared to the baseline model.

Figure 7 compares the performance of Gemini Pro and GPT-3.5, both without reasoning frameworks, with Flan-T5 and other experimented PLMs. Gemini Pro achieved the highest F1 score of 0.65, followed closely by GPT-3.5 with an F1 score of 0.62. Flan-T5 performed moderately with an F1 score of 0.57, while BioLinkBERT, SciBERT, ClinicalBERT, and CTBioBERT displayed lower F1 scores ranging from 0.46 to 0.53.

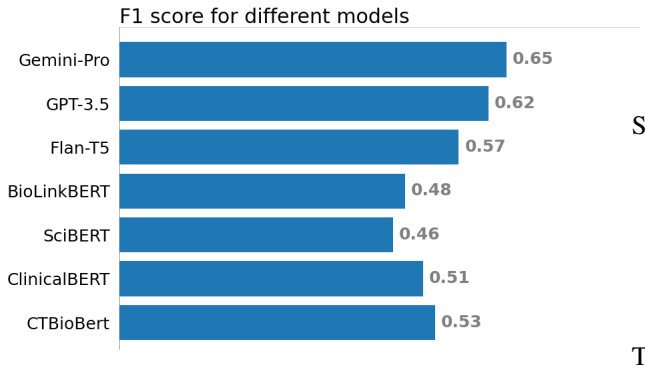


Figure 7: Performance (F1 Score) of the different experimented models. Note: CTBioBert represents the model ClinicalTrialBioBert-NLI4CT.

A comparative analysis between GPT-3.5 and Gemini Pro highlighted GPT-3.5's shortcomings in tasks requiring logical reasoning. Appendix A presents the example responses for both the models. The appendix further analyzes potential reasoning errors made by GPT-3.5 and Gemini Pro.

6 Conclusion

This paper presents an evaluation of several pretrained language models (PLMs), and GPT-3.5, Gemini Pro, under zero-shot conditions. Our analysis focuses on assessing the reasoning capabilities of GPT-3.5 and Gemini Pro and their adeptness at logical problem-solving. In the NLI4CT task, we achieved an F1 score of 0.691, consistency of 0.71, and faithfulness of 0.90. Additionally, our findings underscore that prompt engineering is crucial for large language models (LLMs). We have made our instruction templates and code publicly available to facilitate reproducibility.

7 Acknowledgments

We would like to thank the anonymous reviewers for their careful reading of our manuscript and their insightful comments and constructive criticism. Their feedback has greatly helped us to improve the clarity and rigor of this work.

References

Hilda Bastian, Paul Glasziou, and Iain Chalmers. 2010. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9):e1000326.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Jay DeYoung, Eric Lehman, Ben Nye, Iain J Marshall, and Byron C Wallace. 2020. Evidence inference 2.0: More data, better models. *arXiv preprint arXiv:2005.04177*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama ai research at semeval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jieyi Long. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Anastasios Nentidis, Konstantinos Bougiatiotis, Anastasia Krithara, and Georgios Paliouras. 2020. Results of the seventh edition of the bioasq challenge. In *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II*, pages 553–568. Springer.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597*.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*.

Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. 2020. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1):17.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. *arXiv preprint arXiv:1806.09102*.

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. Thifly research at semeval-2023 task 7: A multi-granularity system for ctr-based textual entailment and evidence retrieval. *arXiv preprint arXiv:2306.01245*.

Appendix

A Example Responses

As illustrated in Figure 8, GPT-3.5 exhibits limitations in numerical reasoning tasks. Although steps 1 and 2 of the example are executed correctly, GPT-3.5 starts hallucinating in step 3, leading to illogical reasoning and ultimately yielding an incorrect conclusion that the statement is incorrect. When the same input is given to Gemini Pro, it correctly concludes that the statement entails the CTR as shown in Figure 9.

Step 1: Assess the exclusion criteria for the primary trial.
- The exclusion criteria mention a Karnofsky Performance status <90% as an exclusion criterion.
- This means that patients with a Karnofsky status below 90% would not be eligible for the trial.

Step 2: Compare the Karnofsky status of 94% mentioned in the hypothesis with the eligibility criteria.
- According to the hypothesis, patients with a Karnofsky status of 94% would be eligible for the primary trial.

Step 3: However, based on the exclusion criteria, a Karnofsky Performance status < 90% is not eligible, regardless of the specific value (i.e. 94%).

Conclusion: *"Patients with a Karnofsky status of 94% are eligible for the primary trial"* is incorrect.

Figure 8: Example of an incorrect response by GPT-3.5

Step 1: Check the inclusion criteria for Karnofsky Performance status.
Step 2: The inclusion criteria state that Karnofsky Performance status must be >= 90%.

Conclusion: *Patients with a Karnofsky status of 94% are eligible for the primary trial* because the inclusion criteria state that Karnofsky Performance status must be >= 90%.

Figure 9: Response by Gemini Pro for the same statement

Figure 10 showcases an excellent example of Gemini Pro's mathematical reasoning.

Hypothesis: Neutropenia affected the majority of patients in cohort 1 of the primary trial.
Label: Contradiction

Explanation:
...
From the context, we know that the total number of patients in cohort 1 of the primary trial is 69, and the number of patients who experienced neutropenia in cohort 1 of the primary trial is 4. Therefore, the percentage of patients in cohort 1 of the primary trial who experienced neutropenia is **4/69 * 100 = 5.8%**.

Since 5.8% is not the majority, the hypothesis is incorrect.

Figure 10: An example response by Gemini Pro showcasing its mathematical reasoning ability.

PEAR at SemEval-2024 Task 1: Pair Encoding with Augmented Re-sampling for Semantic Textual Relatedness

Tollef Emil Jørgensen
Norwegian University of Science and Technology
tollef.jorgensen@ntnu.no

Abstract

This paper describes a system submitted to the supervised track (Track A) at SemEval-24: *Semantic Textual Relatedness for African and Asian Languages*. Challenged with datasets of varying sizes, some as small as 800 samples, we observe that the PEAR system, using smaller pre-trained masked language models to process sentence pairs (**P**air **E**ncoding), results in models that efficiently adapt to the task. In addition to the simplistic modeling approach, we experiment with hyperparameter optimization and data expansion from the provided training sets using multilingual bi-encoders, sampling a dynamic number of nearest neighbors (**A**ugmented **Re**-sampling). The final models are lightweight, allowing fast experimentation and integration of new languages.

1 Introduction

The overall aim of the Semantic Textual Relatedness (STR) shared task (Ousidhoum et al., 2024b) is to correctly predict the relatedness between a given sentence pair on a scale from 0 to 1, described as closeness in meaning (Abdalla et al., 2023; Ousidhoum et al., 2024a), exemplified by *expressing the same views* and *one elaborating on the other*. This shared task covers a broader aspect of the well-established semantic textual similarity (STS) field, which fails to address the intuitive relatedness between two sentences.

From available STS data, such as from the SemEval-2012 task on similarity (Agirre et al., 2012), the sentences "A man is peeling a banana" and "A woman is peeling a potato" receive a normalized similarity of 0.3. In contrast, the two descriptions have a higher degree of relatedness, where *something is being peeled*. Relatedness tends to focus less on equivalence and paraphrasing and more on the broader case of entailment and the cause-effect relationship between two sentences. The task consists of three tracks: A (supervised), B (unsupervised), and C (cross-lingual). The system described here will only consider Track A, allowing the use of any training data. Refer to Ousidhoum et al. (2024a) for more details.

The System and Constraints This paper proposes a system for any language with an available pre-trained masked language model (MLM), such as BERT or RoBERTa, used to process pairs of sentences with full cross-attention. The constraint of using limited-size MLMs was set early in the project to study their performance compared to the impressive baselines observed through existing multilingual bi-encoders. However, following ideas of Thakur et al. (2021), the addition of weakly supervised labels from such bi-encoders was added as an optional step to inspect its impact on smaller datasets.

Being unfamiliar with most of the involved languages and thus being unable to verify the results, no language-specific rules were implemented. Consequently, no text manipulation (such as paraphrasing and replacing words), back-translation, or normalization steps were applied. While the task organizers permitted the use of any available data for the supervised track, in addition to large language models to a limited extent, the presented approach only uses the supplied training dataset per language. While performance suffers in some cases, we hope that the aforementioned constraints help to support as many future languages as possible with little to no modification. Continuing the idea of supporting lower-resourced languages, this system only uses *base* size transformer MLMs, ranging from 110M to 125M parameters.
All code is available on GitHub.[1]

2 Data

The full dataset for SemRel consists of 14 languages. However, only 9 of the 14 languages

[1] https://github.com/tollefj/SemRel-2024

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1405–1411

Language	ISO 639-2/3	Family	Selected Model	Train	Dev	Test	Total
Amharic	amh	Afro-Asiatic	Davlan/xlm-roberta-base-finetuned-amharic	992	95	171	1,258
Algerian Arabic	arq	Afro-Asiatic	CAMeL-Lab/bert-base-arabic-camelbert-da	1,262	92	584	1,938
Moroccan Arabic	ary	Afro-Asiatic	CAMeL-Lab/bert-base-arabic-camelbert-da	925	70	427	1,422
Hausa	hau	Afro-Asiatic	Davlan/xlm-roberta-base-finetuned-hausa	1,763	212	603	2,578
English	eng	Indo-European	FacebookAI/roberta-base	5,500	250	2,500	8,250
Spanish	esp	Indo-European	PlanTL-GOB-ES/roberta-base-bne	1,562	140	600	2,299
Marathi	mar	Indo-European	l3cube-pune/marathi-roberta	1,155	293	298	1,746
Kinyarwanda	kin	Niger-Congo	Davlan/xlm-roberta-base-finetuned-kinyarwanda	778	102	222	1,102
Telugu	tel	Dravidian	l3cube-pune/telugu-bert	1,146	130	297	1,573

Table 1: Included languages and their respective families, along with data sources and data split size.

are included for Track A and have labeled relatedness scores between 0 and 1. Table 1 contains an overview of the languages, data sizes, and selected language models for experiments. Besides the differences in data size, the score distributions also vary greatly, as evident from the four examples in Figure 1. Moreover, when inspecting the

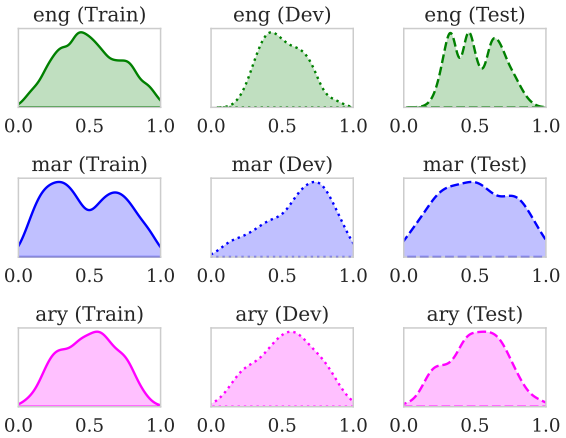


Figure 1: Examples of score distributions.

textual distributions through adversarial validation, modeled by adding a binary classification head to an XLM-R model (Conneau et al., 2020), most languages were seemingly sampled from the same distribution, with an expected ROC-AUC score of 0.5.[2] The English test split, however, had distributions deviating from the train split, shown in Figure 2. ROC curves for more languages are found in Appendix A. Attempts were made to iteratively sample the training set until a better distributional match with the test set was found, with little success in improving results. The more data, the better.

3 Related Work

Semantic Textual Relatedness (STR), in the context of language modeling and prediction, has considerably less research connected to it than Semantic Textual Similarity (STS), which has several datasets and evaluation benchmarks openly available (Muennighoff et al., 2023), many of which tied to the STS task within SemEval (Agirre et al., 2012). The included data is mainly monolingual English, but more recent additions added limited multilingual and cross-lingual tasks (Cer et al., 2017; Chen et al., 2022). The first STR dataset was introduced by (Abdalla et al., 2023), including a monolingual dataset of 5,500 English sentence pairs. New for this task is the inclusion of several low-resource languages not yet studied at the sentence level.

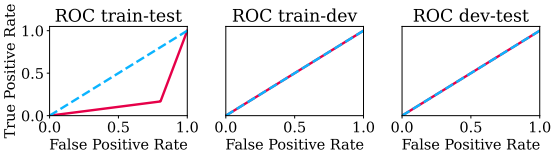


Figure 2: ROC curves for train/dev/test splits on the English data.

The field of natural language processing has drastically changed since the release of the majority of the datasets and shared tasks for semantic textual similarity, where the top-scoring methods typically included a significant amount of feature engineering based on methods like n-gram overlaps, edit distance, and longest common substrings, word alignments, and more, applied to both regression and deep learning models (Tian et al., 2017; Maharjan et al., 2017). Additionally, knowledge-informed systems included semantic information with WordNet and word frequency corpora (Wu et al., 2017). Applying the same efforts to new languages would require significant work, such as collecting new corpora.

Sentence Embeddings Modeling similarity between sentences is commonly associated with *sentence embedding* models, some of which include

[2] An ROC-AUC score of 0.5 indicates that a model cannot differ between samples in the provided data sources.

more than a billion gathered training pairs (Reimers and Gurevych, 2019; Wang et al., 2024). While applicable across many languages and domains, with models initialized from the XLM-Roberta models (Conneau et al., 2020), the included languages do not cover many of which are part of SemRel 2024.

Encoding Sentence Pairs This paper focuses on sentence-pair modeling, encoding the sentences with existing pre-trained language models to create a simpler model that allows fast and easy implementation for any language. This modeling scheme, referred to as cross-encoders, indicating full (cross) self-attention over the entire context, is well explained in previous work by (Wolf et al., 2019; Vig and Ramea, 2019; Humeau et al., 2020). Furthermore, cross-encoders have succeeded in supervised and unsupervised applications (Thakur et al., 2021; Liu et al., 2022). In addition to the sentence scoring, we follow the work by Thakur et al. (2021) to augment data with a bi-encoder, although on much smaller datasets, where the original work was carried out on data up to millions of samples. For details on bi-encoders and sentence embedding models, refer to the excellent implementations by (Reimers and Gurevych, 2019; Humeau et al., 2020; Liu et al., 2022). The baseline provided by the task organizers is LaBSE, a dual-encoder BERT-based sentence embedding model (Feng et al., 2022).

4 System Overview

After restructuring the provided datasets into sentence pairs with their respective labels, they are passed to a MLM with an added regression head; using a sigmoid layer on top of the pooled output, the model is trained using a single-class binary cross-entropy loss, with mean reduction:

$$\ell(x, y) = \frac{1}{n}\sum_{i=1}^{n}\{l_1, \ldots, l_N\}^\top$$
$$l_n = -w_n \left[y_n \log \sigma(x_n) + (1 - y_n) \log(1 - \sigma(x_n))\right]$$

The models (Table 1) were chosen based on searches for existing models in the tasks' languages and closely related language families. In the development phase, scoring was based on 5-fold validation, benchmarked with language-specific and merged data. Experiments, including those presented in Section 6, are on the final release of labeled test datasets.

Augmented Re-sampling In a separate module, a bi-encoder (*multilingual-e5-base*) is employed to find the closest non-existing sentence pairs in the data by creating sentence embeddings and searching for nearest k neighbors with cosine similarity. Before initializing the bi-encoder, the cross-encoder is trained for $\mathcal{E}_{weak}$ epochs before predicting weak labels for the augmented pairs $(s_i, s_j, pred_{i,j})$, which are added to the training data. k determines the number of nearest neighbors to retrieve for each source sentence. A Figure outlining the weak supervision pipeline is found in Figure 3.

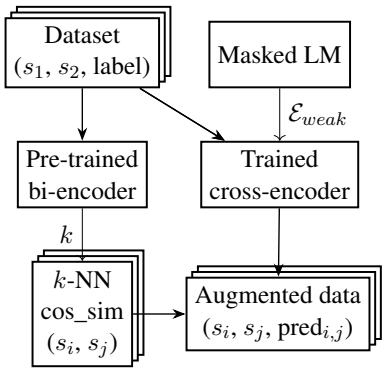


Figure 3: The weak supervision pipeline

Modularity A big focus in the development was to keep it as modular as possible. Models, parameters, data selection, and more are easily controlled through passed arguments. Furthermore, the cross-encoder is provided as a standalone module with varying levels of abstraction, e.g., calling *fit* directly or through a provided training pipeline, including optional weak supervision labeling.

5 Experimental Setup

All experiments and evaluations use the official train/dev/test data splits where applicable, and scores are presented by the Spearman rank correlation coefficient multiplied by 100. In the development phase, we studied the effect of combining or using only per-language data, working as an initial baseline before dev- and test labels were released. This was done by 5-fold validation. As stated in Section 1, no text manipulation or preprocessing was done to keep evaluations fair across languages. Moreover, upon manual inspection, the data seemed sufficiently preprocessed. The following definitions will be used to differ between model configurations:

- **`init`**: no training, only initial weights
- **`all`**: trained on all languages combined
- **`lang`**: trained on one language

Experiments were conducted in three parts:

1. Multilingual sentence embeddings with multilingual-e5-base (Wang et al., 2024)
2. Multi+monolingual MLMs as cross-encoders with XLM-R (Conneau et al., 2020) and models from Table 1.
3. Augmented data from bi-encoders

Augmentation and optimization As the data sizes and model configurations vary, Optuna (Akiba et al., 2019) is set up to search for parameter values for learning rates, k, epochs ($\mathcal{E}$), weak training epochs ($\mathcal{E}_{weak}$), and max gradient norm ($\mathcal{G}$) for clipping. With the augmentation being highly experimental for smaller datasets, we refrain from modifying the bi-encoder and use only its initial weights. Thus, this part of the system can easily be swapped with future models. While limiting the search, the learning rate, gradient clipping, and the k nearest neighbors for augmentation proved to be the most crucial parameters. Table 2 lists the parameters and ranges.

Hparam	Type	Search Space
lr	float	10^{-6} to 10^{-4} (log)
k	int	0 to 3
$\mathcal{E}$	int	1 to 5
$\mathcal{E}_{weak}$	int	0 to 2
$\mathcal{G}$	float	0.1 to 1.0

Table 2: Hyperparameter search space.

No External Data Given the readily available sentence similarity data, such as the STS-Benchmark dataset (Cer et al., 2017), experiments were done to include it in the training pipelines. However, we observed no benefits from this, likely affected by the diverging definitions and annotation styles of relatedness and similarity. Figure 4 shows scores with and without adding the STS-Benchmark dataset (Cer et al., 2017).

6 Results

Results from k-fold validation to quantify the differences between combining training sets vs. training per language show that combining data has a clear benefit. See table 3. However, important factors to consider are data size (e.g., 12,000 vs. 600 samples) and that we are validating in-domain. However,

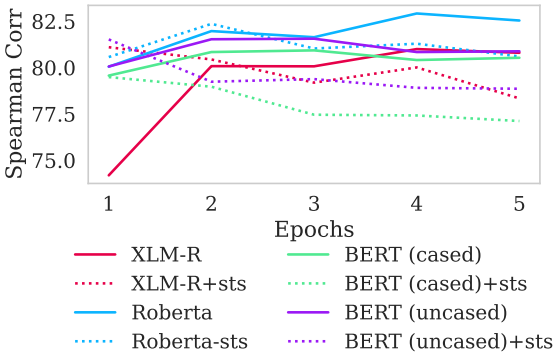


Figure 4: Scores on English-dev with and without STS data (dotted lines)

similar results on the development set show diminishing effects for merging data when predicting out-of-domain data. Development set scores are found in Appendix B. Furthermore, the results indicate how powerful MLMs are once trained, outscoring the *e5* model for most languages. Although the system performed well on the smaller dev sets, ranking in the top 2-4 in most languages in the competition, it struggled on the test sets. This is likely attributed to data distribution, overfitting, and thus failure to generalize relatedness. From the results with augmentation in Table 5, the observed change from default parameters is marginal for most languages. Increasing k without parameter optimization resulted in strictly negative results. Scores on the test set, including top scores and the LaBSE baseline, are shown in Table 4. Despite the lackluster improvement from augmentation and data expansion, the modeling scheme is still promising, outperforming the baseline (in official submissions) for 6/9 languages and 8/9 for the rerun without any changes to optimization configurations.

7 Conclusion

After testing the capabilities of commonly used models for masked language modeling and sentence embeddings, we find MLMs efficient at distinguishing relatedness with little training data. Although attempts at optimizing parameters for in-distribution data resulted in little to no performance gains, there are likely better-suited augmentation strategies for further improving performance with as little source data as possible. As a closing remark, we hope that the provided system may serve as a valuable tool for future developments in semantic textual relatedness.

	arq	amh	eng	hau	kin	mar	ary	esp	tel
XLM init	-2.64 (5.07)	-10.75 (7.67)	-15.49 (3.33)	-4.18 (6.07)	1.03 (3.90)	-7.65 (7.49)	-18.98 (4.88)	0.80 (4.94)	-11.98 (9.32)
XLM all	58.23 (5.30)	**84.56** (1.53)	83.63 (1.36)	**72.25** (0.66)	59.70 (4.09)	**83.44** (2.56)	**82.01** (3.04)	64.73 (3.23)	77.96 (3.87)
XLM lang	39.03 (4.49)	73.22 (3.02)	83.27 (0.89)	63.57 (1.96)	31.40 (7.34)	74.31 (3.02)	69.14 (4.16)	58.72 (8.00)	71.40 (3.99)
e5 init	50.41 (2.82)	75.86 (1.88)	80.72 (0.87)	52.38 (1.93)	46.20 (5.30)	77.00 (1.33)	36.03 (1.59)	60.30 (1.40)	75.28 (1.49)
e5 all	59.45 (2.34)	84.52 (0.88)	86.43 (0.55)	69.01 (0.19)	**69.08** (3.43)	**84.62** (1.35)	81.20 (1.44)	67.16 (2.44)	**80.14** (0.97)
e5 lang	**59.50** (3.25)	82.27 (2.35)	**86.72** (1.02)	68.43 (2.10)	63.04 (3.56)	82.89 (0.32)	75.73 (1.02)	**67.21** (0.39)	77.94 (1.23)

Table 3: 5-fold validation from training datasets using *multilingual-e5-base* (bi-encoder) and *XLM-Roberta base* (cross-encoder). Scores are the average correlation with standard deviations. Bold: best scores per language.

Model/Language	Multiling	k	arq	amh	eng	hau	kin	mar	ary	esp	tel
Baseline (LaBSE)	y	0	60.00	85.00	83.00	69.00	72.00	88.00	77.00	70.00	82.00
Best result	-	-	**68.23**	**88.86**	**85.96**	**76.43**	**81.69**	**91.09**	**86.26**	**74.04**	**87.34**
e5 init (multiling)	y	0	45.32	72.56	80.39	51.23	51.38	77.37	40.14	58.75	77.43
e5 all (multiling)	y	0	59.28	82.06	83.53	68.36	71.61	87.27	78.27	69.16	83.25
e5 lang (multiling)	y	0	60.68	81.46	83.55	69.97	71.87	87.91	77.75	69.02	82.24
e5 init	n	0	43.94	9.02	82.69	40.79	48.23	52.76	15.41	65.22	28.69
e5 all	n	0	59.32	14.48	82.88	61.87	68.15	69.59	77.30	71.26	43.64
e5 lang	n	0	55.30	13.70	83.54	63.63	63.60	67.88	36.11	70.86	34.21
xlm-r init	y	0	-1.10	12.45	-4.23	-0.75	1.93	-10.24	-28.12	1.73	-14.68
xlm-r all	y	0	59.88	83.42	83.69	70.74	67.48	85.99	83.04	71.39	85.75
xlm-r lang	y	0	47.66	81.90	83.46	70.17	56.76	85.84	82.23	69.73	80.78
custom init	n	0	-10.97	20.40	10.09	9.52	14.35	-3.35	-1.91	-3.35	8.40
custom lang	n	0	40.04	83.86	83.31	68.79	72.09	86.10	81.15	72.05	83.46
custom lang	n	1	44.56	81.99	83.42	66.56	72.75	85.83	80.74	71.95	84.42
custom lang	n	2	43.75	81.89	83.27	65.66	70.27	85.72	80.49	71.79	85.05
custom lang	n	3	42.28	81.13	83.39	64.67	71.47	85.36	80.37	72.29	84.73
PEAR$_{test}$	n	+	46.33	83.42	84.79	69.41	77.22	85.60	81.53	71.01	82.75
PEAR$_{rerun}$	n	+	48.58	85.72	83.95	70.68	73.92	88.81	81.68	72.52	86.82

Table 4: Performance on the test set, ordered by languages as presented on the task website. *Multiling* indicates whether the model was pre-trained on multilingual data. k indicates the k-NN resamples used (+: different k per language). *custom*: monolingual models as listed in Table 1. Bold: best score (from all submissions to Track A). Underline: second best from the experiments.

lang	lr	k	$\mathcal{E}$	$\mathcal{E}_{weak}$	$\mathcal{G}$	score	Δ
arq	9.80e-5	1	4	2	0.82	48.58	+8.54
amh	8.42e-5	3	5	2	0.80	85.72	+1.86
eng	3.34e-5	2	2	2	0.12	83.95	+0.64
hau	4.87e-5	0	3	2	0.65	70.68	+1.89
kin	2.47e-5	0	5	1	0.67	73.92	+1.83
mar	5.32e-5	3	2	2	0.42	88.81	+2.71
ary	9.01e-5	1	4	2	0.99	81.68	+0.53
esp	2.40e-5	1	5	2	0.85	72.52	+0.47
tel	3.66e-5	3	3	1	0.66	86.82	+3.36

Table 5: Parameters found from the search space in Table 2. Δ indicates change vs. default parameters.

8 Limitations

Few models were tested per language for the competition. Alternative multi- and monolingual models could provide much better results, especially for Algerian Arabic. This limitation is also influenced by the lack of understanding of most involved languages, e.g., to inspect the source datasets used for pretraining. Finally, grouping specific languages for training, such as merging Indo-European and Afro-Asiatic languages, was not explored.

9 Ethical Considerations

The final system performs predictions of input texts. Predictions may impose ethical concerns, e.g., when used for public-facing applications. Furthermore, automating *relatedness* has possible side effects in bias and fairness towards specific nationalities. For further details about the data and annotation, refer to Ousidhoum et al. (2024a).

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1094–1106, Seattle, United States. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring.

Fangyu Liu, Yunlong Jiao, Jordan Massiah, Emine Yilmaz, and Serhii Havrylov. 2022. Trans-encoder: Unsupervised sentence-pair modelling through self- and mutual-distillations.

Nabin Maharjan, Rajendra Banjade, Dipesh Gautam, Lasang J. Tamang, and Vasile Rus. 2017. DT_Team at SemEval-2017 task 1: Semantic similarity using alignments, sentence-level embeddings and Gaussian mixture model output. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 120–124, Vancouver, Canada. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks.

Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 task 1: Leverage kernel-based traditional NLP features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 191–197, Vancouver, Canada. Association for Computational Linguistics.

Jesse Vig and Kalai Ramea. 2019. Comparison of transfer-learning approaches for response selection in multi-turn conversations. In *Workshop on DSTC7*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents.

Hao Wu, Heyan Huang, Ping Jian, Yuhang Guo, and Chao Su. 2017. BIT at SemEval-2017 task 1: Using semantic information space to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 77–84, Vancouver, Canada. Association for Computational Linguistics.

A Adversarial Validation

Figures 5 and 6 show the ROC curve for a selection of languages where the AUC value deviated from the norm. English was an outlier here, where the test set is seemingly out-of-distribution. An XLM-Roberta base model set up as a cross-encoder was used for classification. 5 epochs, learning rate 2×10^{-5}. All languages not shown in the figures have an expected ROC-AUC close to 0.5.

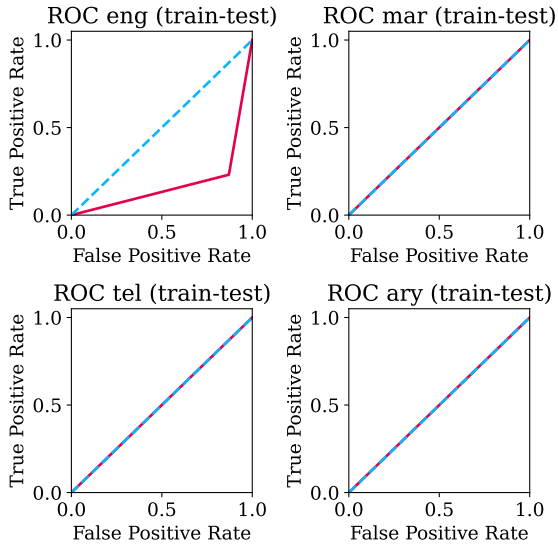


Figure 5: Adversarial validation for Train vs Test. English (eng), Marathi (mar), Telugu (tel) and Moroccan Arabic (ary).

B Development Set Results

Table 6 shows the results on dev sets for *multilingual-e5-base*, *XLM-Roberta-base*, and language-specific masked language models (as defined in Table 1). Modeling configurations are the same as listed in Section 5 – repeated below:

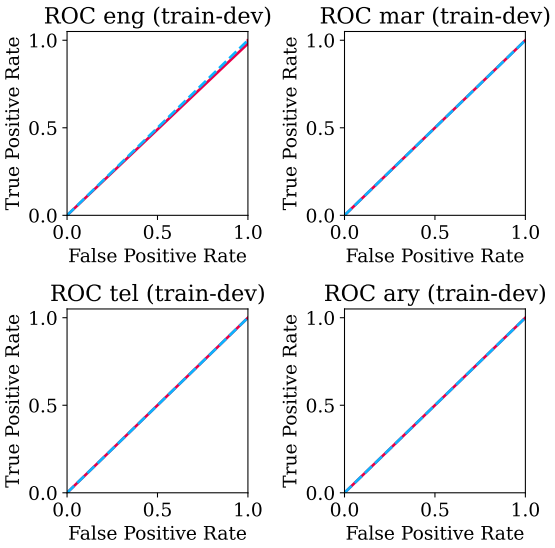


Figure 6: Adversarial validation for Train vs Dev. English (eng), Marathi (mar), Telugu (tel) and Moroccan Arabic (ary).

- **`init`**: no training, only initial weights
- **`all`**: trained on all languages combined
- **`lang`**: trained on one language

Lang	e5 multilingual			XLM-Roberta			MLM	
	init	all	lang	init	all	lang	init	lang
arq	39.70	54.26	**59.71**	-11.11	59.22	57.32	4.76	38.90
amh	61.82	78.47	77.79	-5.30	**86.57**	83.38	19.65	85.90
eng	78.31	81.44	82.10	-12.06	80.88	81.05	10.98	**82.79**
hau	45.01	73.27	72.91	-9.19	**76.39**	75.41	12.95	75.99
kin	27.80	62.55	**65.63**	-18.13	59.90	48.67	7.09	64.73
mar	72.48	81.56	80.56	-15.13	84.24	82.86	-9.91	**84.73**
ary	44.21	78.84	73.79	-28.55	**83.96**	82.61	-19.75	81.43
esp	62.63	68.54	63.16	22.38	**71.24**	65.01	12.80	68.23
tel	77.35	**82.27**	79.75	-16.67	80.34	80.57	18.36	80.74

Table 6: Results on the development sets. Bold: best score per language.

BCAmirs at SemEval-2024 Task 4: Beyond Words: A Multimodal and Multilingual Exploration of Persuasion in Memes

Amirhossein Abaskohi[†*], **Amirhossein Dabiriaghdam**[‡*], **Lele Wang**[‡], **Giuseppe Carenini**[†]
[†]Department of Computer Science, University of British Columbia
[‡]Department of Electrical and Computer Engineering, University of British Columbia
Vancouver, Canada
{aabaskoh, carenini}@cs.ubc.ca
{amirhossein, lelewang}@ece.ubc.ca

Abstract

Memes, combining text and images, frequently use metaphors to convey persuasive messages, shaping public opinion. Motivated by this, our team engaged in SemEval-2024 Task 4, a hierarchical multi-label classification task designed to identify rhetorical and psychological persuasion techniques embedded within memes. To tackle this problem, we introduced a caption generation step to assess the modality gap and the impact of additional semantic information from images, which improved our result. Our best model utilizes GPT-4 generated captions alongside meme text to fine-tune RoBERTa as the text encoder and CLIP as the image encoder. It outperforms the baseline by a large margin in all 12 subtasks. In particular, it ranked in top-3 across all languages in Subtask 2a, and top-4 in Subtask 2b, demonstrating quantitatively strong performance. The improvement achieved by the introduced intermediate step is likely attributable to the metaphorical essence of images that challenges visual encoders. This highlights the potential for improving abstract visual semantics encoding.[1]

1 Introduction

In this digital age, the influence of persuasive techniques, particularly in memes, is a key focus. Propaganda, using various psychological techniques, shapes information for specific agendas. Research on political memes, such as Kulkarni (2017)'s work, emphasizes their role in communication and satire. Another study on COVID-19 memes (Wasike, 2022) underscores the importance of expert-sourced, objective memes in influencing public opinion and aiding public health campaigns. As a result, understanding persuasive techniques in memes within disinformation campaigns is crucial for grasping their impact on public perception and discourse. These campaigns usually succeed in influencing users by employing various rhetorical and psychological strategies in memes, including but not limited to *causal oversimplification*, *thought-terminating cliché*, and *smear* techniques.

To address this concern, we participated in the SemEval-2024 shared task 4, as outlined by (Dimitrov et al., 2024). The primary objective of this shared task is to develop models specifically designed to detect rhetorical and psychological techniques within memes. In summary, this task involves three subtasks. In Subtask 1 the input is the textual content of a meme only. This could include any written information present within the meme, and the goal is to identify one of the 20 persuasion techniques present in the meme's textual content. The identification is based on a hierarchical structure, and the techniques are organized in a tree-like fashion. Subtask 2a involves both textual and visual content analysis of memes, and information present in both the written content and the visual elements of the meme are considered. The task is to identify the presence of 22 persuasion techniques, utilizing a hierarchical structure similar to Subtask 1. Subtask 2b is a binary classification version of Subtask 2a. The training set released for all subtasks contains only English memes. However, alongside the English language, the test datasets contain memes in three low-resource languages (Arabic, Bulgarian, and North Macedonian) that aim to evaluate the zero-shot capability of the proposed models.

Although we participated in all subtasks, we specifically focused on Subtask 2 which uses both the textual and visual modality of memes to do a multi-label classification. To achieve better results, we introduced an intermediate step, the meme captioning step. Subsequently, we employed these generated captions to compare the performance of different models like LLaVA-1.5 (Liu et al., 2023), Vicuna-1.5 (Chiang et al., 2023), BERT (Devlin

[*] Equal Contribution
[1] Our code is publicly available at https://github.com/AmirAbaskohi/Beyond-Words-A-Multimodal-Exploration-of-Persuasion-in-Memes.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1412–1423

Subtask	Ours	Baseline	Rank
2a - English	**70.497**	44.706	3
2a - Bulgarian	**62.693**	50.000	1
2a - North Macedonian	**63.681**	55.525	1
2a - Arabic	**52.613**	48.649	1
2b - English	**80.337**	25.000	4
2b - Bulgarian	**64.719**	16.667	4
2b - North Macedonian	**64.719**	09.091	4
2b - Arabic	**61.487**	22.705	1
1 - English	**69.857**	36.865	2
1 - Bulgarian	**44.834**	28.377	13
1 - North Macedonian	**39.298**	30.692	12
1 - Arabic	**39.625**	35.897	9

Table 1: Results of our best model (at the time of submitting evaluation results) on the test dataset of different subtasks. In the table the ranking and the values are based on hierarchical F1.

et al., 2018), and RoBERTa (Liu et al., 2019). This comparative analysis aimed to elucidate the role of the memes' text, the generated captions, and the memes' images in understanding persuasion techniques used in memes. The results of our best model (at the time of submitting evaluation results) that uses RoBERTa and our ranking relative to other teams in different subtasks are summarized in Table 1. It can be seen that our method performed well in the Subtask 2 (our main focus) for all four languages, and also the English subset of the first subtask. Our model struggled with non-English subsets of Subtask 1 since (I) we did not have access to the image of the meme and therefore no caption was available, and (II) our models only understood English, so we relied on a translation (using Google Translate[2]) of the memes' text.

Prior approaches have tried to narrow the gap between visual and textual realms to enhance image captioning. However, these methods primarily emphasized captioning visual details through textually enriched image features, rather than delving into the metaphorical significance inherent in images, particularly in the context of memes. To the best of our knowledge, this is the first work that focuses on the metaphorical semantic gap in multimodal language models to examine the gap between image and text modalities. Our ultimate goal was to gain insight into discrepancies between visual and textual metaphors in these systems. In summary, our contributions are twofold: (I) Addressing the classification problem of persuasion techniques in memes using multimodal models, and (II) Investigating the modality gap between textual and image components in multimodal models.

[2] https://translate.google.com

This paper is organized as follows: In Section 2, we review prior research on hierarchical classification, persuasion techniques classification from memes, and the gap between textual and visual modalities. Section 3 introduces the datasets, discusses models, and outlines our approach for hierarchical persuasion technique classification. Section 4 presents and discusses our experiments and findings. Finally, in Section 5, we conclude our work, summarizing key contributions and suggesting directions for future research.

2 Background

Modality Gap. In researching the modality gap between modalities, Zhao et al. (2023) presents ChatBridge, a novel multimodal large language model (MLLM) that employs language as a catalyst to bridge the gap between various modalities, such as text, image, video, and audio. By leveraging the expressive capabilities of language, ChatBridge connects different modalities using only language-paired bimodal data, showcasing strong quantitative and qualitative results on zero-shot multimodal tasks. Furthermore, Chen et al. (2023) addresses the limitations of existing MLLMs in effectively extracting and reasoning visual knowledge. The proposed model, LION, injects dual-level visual knowledge, incorporating fine-grained spatial-aware visual knowledge and high-level semantic visual evidence. LION outperforms existing models in vision-language tasks, including image captioning, visual question answering, and visual grounding, through a two-stage training process. By extending these insights to the unique realm of memes, our work not only adds to the growing body of research on multimodal models but also sheds light on the gap between visual and textual modalities, especially in the metaphorical landscape.

Persuasion Technique Classification. In exploring persuasion techniques in texts and images, Dimitrov et al. (2021) present a comprehensive framework for meme analysis. The study defines 22 techniques and provides an annotated dataset for conducting nuanced examinations of textual and multimodal memes. The incorporation of historical and mythological references adds depth to understanding the challenges in this domain. Moreover, in the work by Messina et al. (2021), the authors introduce transformer-based (Vaswani et al., 2017) models, VTTE and DVTT, for processing textual and visual content in memes. These models effec-

tively identify persuasion techniques, with DVTT showing superior performance, particularly in fine-tuning feature extractors. Given the prevalence of Large Language Models (LLMs) employing similar architectures as DVTT, our experiments include utilizing LLMs and MLLMs, which we explain in Section 3, to further investigate and advance the detection of persuasion techniques in memes.

3 Methodology

Building upon prior research on MLLMs, the prevailing method involves tokenizing image concepts and conveying these tokens alongside textual tokens to a language model. While these models possess the ability to impart more semantic information from the image, their focus typically centers on identifying objects and their relationships within the image (Park and Paik, 2023). Consequently, this study explores the impact of initially prompting the model to generate descriptive information aimed at conveying semantic context. We utilize this information for data classification, comparing it to the conventional approach of fine-tuning an end-to-end model. In this section, we outline our approach for generating meme captions and subsequently discuss our classifiers.

3.1 Caption Generation

In this paper we used three different models for generating meme captions: BLIP-2 (Li et al., 2023), LLaVA-1.5-7B (Liu et al., 2023), and GPT-4 (OpenAI et al., 2023). We fine-tuned BLIP-2 and LLaVA-1.5 for generating captions and used GPT-4 in zero-shot settings. Based on our results which are explained in Appendix C, we found out that LLaVA-1.5 outperforms BLIP-2 in the quality of generated captions. In order to fine-tune our meme captioning model, we used MemeCap (Hwang and Shwartz, 2023) dataset. Figure 1 illustrates the fine-tuning loop for LLaVA. This involved generating descriptions of memes capturing the conveyed message to uncover deeper layers of semantic understanding. Subsequently, we utilized this fine-tuned LLaVA to generate captions for the persuasion technique datasets. The generated captions provided supplementary data, offering further insights into the memes and enabling us to examine the effects of additional semantic information. Additionally, these captions were utilized to evaluate the modality gap within MLLMs. As elaborated in the subsequent section, we studied the distinctions between incorporating both the meme and its caption during the classification phase.

Considering our results in meme caption generation, discussed in Appendix C, we identified two potential issues with the captions generated by our fine-tuned models. The first issue concerns the domain disparity between the MemeCap and persuasion datasets. The memes in the task's dataset often contain toxic content and usually require a deep understanding of background knowledge and events. The second issue relates to the brevity of captions in the MemeCap dataset, which typically only mentions the meme's final goal. This brevity may not provide sufficient information for detecting persuasion approaches. Consequently, we opted to generate captions using GPT-4 in zero-shot settings. GPT-4, the latest MLLM from OpenAI, was employed in our study utilizing the recently released API known as **gpt-4-vision-preview**. This model exhibits remarkable proficiency across diverse tasks such as visual question answering and image captioning. See Appendix D for details in caption generation with GPT-4.

3.2 Persuasion Technique Classification

After generating captions for the memes, except for Subtask 1 where only the text written in the meme is provided, for classifying persuasion techniques we have three features available: meme, the text written in it, and our generated caption. In order to investigate the effect of our proposed model and assess the modality gap in MLLMs, we evaluate the effect of different combinations of these features. This will be explained further in Section 4. As our classifier model, whether in a multi-label setting like subtasks 1 and 2a, or 2b which is binary classification, we used different families of models from LLMs, MLLMs, and Language Representation Models (LRMs). These models are as follows:

LLMs. Given the promising results of LLMs in semantic classification tasks (Sun et al., 2023; Abaskohi et al., 2023), we opted to use them as our classifiers. To ensure a fair comparison and analyze the modality gap, we utilized the same LLM used in LLaVA, namely Vicuna. We initially fine-tuned the LLM solely with the text written in the meme, followed by fine-tuning with both the text in the meme and the meme's caption.

MLLMs. To assess the impact of employing our intermediate step of generating meme captions and

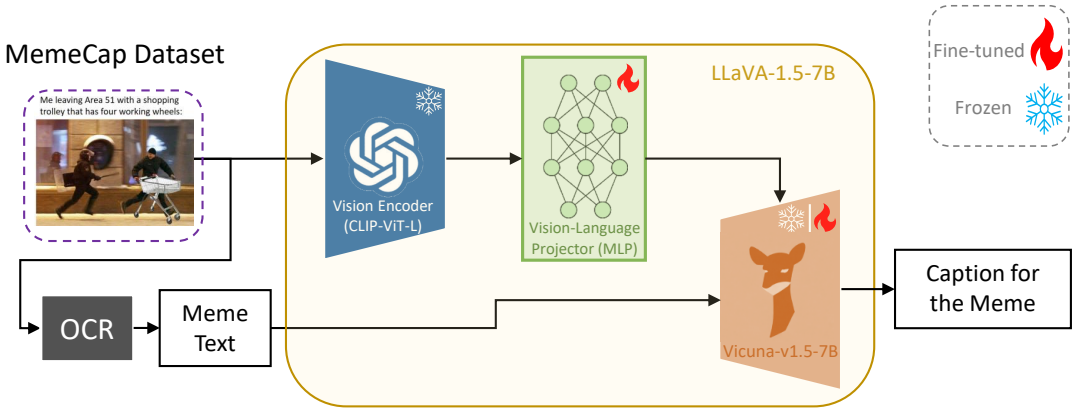


Figure 1: The figure depicts the supervised fine-tuning loop of the LLaVA-1.5-7B model on the MemeCap dataset for caption generation. The OCR module extracts text from the meme images. The vision encoder (CLIP), a frozen component of LLaVA-1.5-7B, processes the meme images. The vision-language projector bridges the gap between CLIP's representation and the embedding space of Vicuna. While CLIP remains frozen, the vision-language projector is fine-tuned. Vicuna component experimented with both frozen and fine-tuned setups to generate captions.

using them alongside the memes, we required training an end-to-end model. To accomplish this, we fine-tuned LLaVA by incorporating the meme's image, the text written within memes with or without the captions associated with them. This augmentation aimed to leverage the additional information conveyed by the captions and potentially enhance the model's performance.

LRMs. In several of our experiments, we employed BERT and RoBERTa as our classifiers. These models utilize only the encoder component of the transformer architecture. Despite the growing dominance of LLMs in various benchmarks, these models remain highly potent, particularly in semantic-related tasks. We fine-tuned the large versions of these models as our classifiers, first solely with the text written in the meme and then with both the text in the meme and the meme's caption.

Multimodal LRMs. After exploring the impact of MLMMs alongside LLMs, we delved into a BERT variant with visual understanding capabilities. Initially, we fine-tuned VisualBERT (Li et al.) on both memes and their accompanying text, with and without captions. However, given the unavailability of a pre-trained large version of VisualBERT we devised a concatenated model comprising RoBERTa-large and a vision encoder, inspired by an example from (Singh et al., 2020). We experimented with CLIP-ViT-large (Radford et al., 2021) as the component for the vision encoder. Subsequently, we concatenated the encoded features from CLIP with encoded features from RoBERTa and employed a linear classifier to determine the class based on the encoded information. We call this model *ConcatRoBERTa* (see Figure 2). To maintain consistency with previous MLLM-based methods, CLIP were frozen during the training phase.

4 Experiments

In this section, we outline our conducted experiments and provide a concise overview of the results obtained. Readers are referred to Appendix A for the details of our experimental settings.

For evaluation of the performance of the models for hierarchical classifications, we use hierarchical-precision, -recall, and -F1 introduced by Kiritchenko et al. (2006). For more information about hierarchical evaluation metrics, see Appendix E.

In the initial set of experiments, we perform hierarchical multilabel classification using the textual content of memes to fine-tune unimodal models (Vicuna, BERT, and RoBERTa) directly for identifying specific persuasion techniques. We compare this approach to multimodal models (LLaVA, VisualBERT, and ConcatRoBERTa) where both textual and visual contents of the meme are provided. Additionally, we conduct a similar experiment with LLaVA model, feeding only the image of the meme without textual data. This comparison aims to assess the information derived from each modality. The motivation behind our decision to compare encoder-only LRMs such as RoBERTa or VisualBERT to larger generative models like Vicuna or LLaVA, is their promising results in classification tasks compared to generative models (Sarrouti et al., 2022; Jiang et al., 2023). During the second stage of our experiments, we proposed to

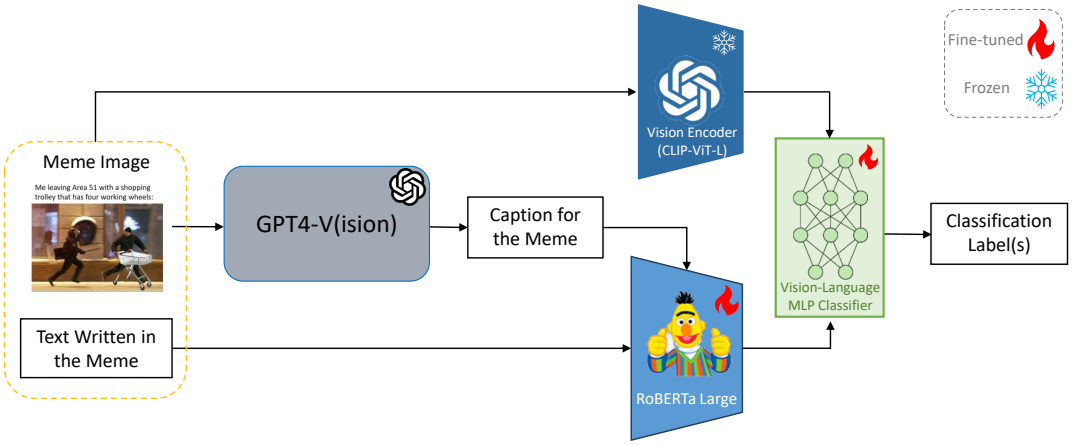


Figure 2: The figure illustrates the architecture of ConcatRoBERTa, our best-performing model. The GPT4-V(ision) component generates a descriptive caption of the meme image. The caption is then combined with the text written in the meme, which is processed by the RoBERTa. The Vision encoder utilizes a pre-trained vision transformer model (CLIP-ViT), to encode and analyze the visual elements of the meme. The MLP Classifier takes the combined visual and textual representations and classifies the meme. RoBERTa and the MLP classifiers are fine-tuned, while CLIP remains frozen.

Model	**H-F1**	**H-Precision**	**H-Recall**
LLaVA-1.5 (image)	58.21	62.74	54.31
LLaVA-1.5 (image+text)	62.59	66.00	59.51
LLaVA-1.5 (image+text+caption from LLaVA-1.5)	63.33	67.02	60.02
Vicuna-1.5 (text)	62.69	71.03	56.10
Vicuna-1.5 (text+caption from LLaVA-1.5)	63.11	70.86	56.88
Vicuna-1.5 (text+caption from GPT-4)	65.337	75.204	57.759
BERT (text)	64.881	75.400	56.938
BERT (text+caption from LLaVA-1.5)	66.455	74.229	60.155
BERT (text+caption from GPT-4)	66.829	75.958	59.659
RoBERTa (text)	66.740	76.846	58.983
RoBERTa (text+caption from LLaVA-1.5)	67.750	73.699	62.690
RoBERTa (text+caption from GPT-4)	69.913	**76.999**	64.021
VisualBERT (image+text)	51.496	39.779	**72.998**
VisualBERT (image+text+caption from LLaVA-1.5)	57.714	57.841	62.690
ConcatRoBERTa (image+text)	65.188	73.443	58.601
ConcatRoBERTa (image+text+caption from LLaVA-1.5)	67.166	75.283	60.629
ConcatRoBERTa (image+text+caption from GPT-4)	**71.115**	76.101	66.742
Baseline	44.706	68.778	33.116

Table 2: Comparison of results of different methods on dev set of Subtask 2a. H-F1, H-Precision, and H-Recall, are hierarchical-F1, -precision, and -recall respectively. As expected, models prefer to receive more information about the image, and models incorporating all features (e.g., text, caption, and image) tend to perform better. However, captions appear to be more informative. This suggests that although some information from the image may not be fully conveyed through text, utilizing models to initially analyze the image, particularly in meme tasks like this, and then prompting them to make decisions based on that analysis, yields superior performance compared to making decisions without leveraging their full capabilities.

create captions for the memes, and subsequently augment the original data with these generated captions. This additional step aims to capture more information from the meme image and adopt this

additional data to improve the results of the hierarchical classification of the memes. In this phase, we mainly focused on subtask 2a. The results of our different methods on the dev set of subtask 2a are presented in Table 2.

From Table 2, we can see the best performing model is ConcatRoBERTa, which has both the image and the text written on the meme as well as the caption generated by GPT-4 as its inputs (Figure 2) with hierarchical F1 score of 71.115 on the dev set of subtask2a. It is worth mentioning that due to time constraints, we could not evaluate test datasets using ConcatRoBERTa by the evaluation deadline, therefore, the submitted results for the test dataset in Table 1 are from RoBERTa model (our second best model). It might be unexpected that MLLMs like LLaVA with text and image of the meme as their input do not perform as well as LLMs like Vicuna with text and caption of the meme in this particular task. This discrepancy could be attributed to the metaphorical nature of memes. Vision encoders, such as CLIP, are primarily trained to comprehend the visual aspects of an image, lacking a focus on the metaphorical meanings embedded in those visual elements. In contrast, language models are more adept at understanding metaphors, given their greater exposure to such linguistic nuances in textual data which has been shown previously (Hwang and Shwartz, 2023). Note the improvement in the results when employing GPT-4 for caption generation instead of LLaVA. As mentioned earlier, it is due to the domain disparity between MemeCap and this task's dataset. Regarding the superiority of the results of fine-tuned LRMs such as RoBERTa compared to LLMs like Vicuna, we argue that LLMs in general tasks are better but often for certain tasks a well-implemented LRM can outperform LLMs. In other words, the performance of LLMs fluctuates significantly based on the limitations of the data and the specific application context. This observation can be attributed to the use of a relatively small generative language model (with only 7B parameters) for a challenging task. Finally, it is not surprising that VisualBERT's results are not as good as other larger models since we only had access to the base version of pre-trained VisualBERT.

Another observation is that by adding an intermediate step of caption generation, results are improved when it is used in a supervised learning manner. In contrast, for the in-context learning scheme (Appendix B), we note that the additional information extracted from memes, specifically captions, did not improve but rather worsened the results. The diverse nature of meme captions, including more details compared to the text within the memes, may misguide the model in focusing on relevant features. In such a setting, the models' in-context learning ability is limited, and giving more information only confuses the model without any gain. Even we tried to use GPT-4 (in a zero-shot setting) for subtask 2b, and its results on the dev set were comparable but worse than using our proposed method (RoBERTa with generated caption from GPT-4), i.e., 73.242 and 79.667 versus 78.434 and 81.333 for macro- and micro-F1, respectively.

5 Conclusion and Future Work

This paper explores the persuasive communication within memes, emphasizing their role in shaping public perception. Through participation in the SemEval-2024 shared task 4, our study delves into the detection of rhetorical and psychological techniques within memes. By employing multimodal models and introducing an intermediate step of meme captioning using LLaVA and GPT-4, we aimed to bridge the gap between textual and visual modalities, thereby enhancing the classification of persuasion techniques. Our experiments demonstrate the effectiveness of this approach, with our best model, ConcatRoBERTa, achieving notable performance improvements. However, we observed that the performance gains varied based on the dataset's nature and the models' sophistication. Nevertheless, our findings contribute to advancing understanding in this domain and pave the way for future research endeavors aimed at combating online disinformation campaigns.

Regarding future work, a deeper analysis into why the model struggles to utilize its image analysis capabilities for classification, despite its proficiency in generating captions (even in zero-shot settings with GPT-4), could be explored through the implementation of chain-of-thought approaches. Additionally, exploring how well the proposed method withstands adversarial attacks is another interesting direction. Adversarial examples, as shown by different studies (Moosavi-Dezfooli et al., 2016; Sadrizadeh et al., 2023; Zhao et al., 2024), have uncovered vulnerabilities in neural models across various tasks. Studying how adding the caption generation step affects the adversarial robustness of our approach compared to end-to-end methods for this task holds promise for future research.

References

Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. 2023. LM-CPPF: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 670–681, Toronto, Canada. Association for Computational Linguistics.

Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. 2023. Lion: Empowering multimodal large language model with dual-level visual knowledge. *arXiv preprint arXiv:2311.11860*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

EunJeong Hwang and Vered Shwartz. 2023. MemeCap: A dataset for captioning and interpreting memes. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1433–1445, Singapore. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*, pages 395–406. Springer.

Anushka Kulkarni. 2017. Internet meme and political discourse: A study on the impact of internet meme as a tool in communicating political satire. *Journal of Content, Community & Communication Amity School of Communication*, 6.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Liunian Harold Li, Mark Yatskar, D Yin, CJ Hsieh, and KW Chang. Visualbert: A simple and performant baseline for vision and language. arxiv 2019. *arXiv preprint arXiv:1908.03557*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Nicola Messina, Fabrizio Falchi, Claudio Gennaro, and Giuseppe Amato. 2021. AIMH at SemEval-2021 task 6: Multimodal classification using an ensemble of transformer models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1020–1026, Online. Association for Computational Linguistics.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Seokmok Park and Joonki Paik. 2023. Refcap: image captioning with referent objects attributes. *Scientific Reports*, 13(1):21577.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Sahar Sadrizadeh, AmirHossein Dabiri Aghdam, Ljiljana Dolamic, and Pascal Frossard. 2023. Targeted adversarial attacks against neural machine translation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Mourad Sarrouti, Carson Tao, and Yoann Mamy Randriamihaja. 2022. Comparing encoder-only and encoder-decoder transformers for relation extraction from biomedical texts: An empirical study on ten benchmark datasets. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 376–382, Dublin, Ireland. Association for Computational Linguistics.

Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2020. Mmf: A multimodal framework for vision and language research. https://github.com/facebookresearch/mmf.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text

classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ben Wasike. 2022. Memes, memes, everywhere, nor any meme to trust: Examining the credibility and persuasiveness of covid-19-related memes. *Journal of Computer-Mediated Communication*, 27(2):zmab024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Man Cheung, and Min Lin. 2024. On evaluating adversarial robustness of large vision-language models. *Advances in Neural Information Processing Systems*, 36.

Zijia Zhao, Longteng Guo, Tongtian Yue, Sihan Chen, Shuai Shao, Xinxin Zhu, Zehuan Yuan, and Jing Liu. 2023. Chatbridge: Bridging modalities with large language model as a language catalyst. *arXiv preprint arXiv:2305.16103*.

A Experimental Settings

All of the experiments were conducted on a Core i9 system with 64GB of RAM and Nvidia RTX3090 GPU with 24GB VRAM.

In all combinations of the experiments in Section 4 involving generative models, the temperature and number of beams for text generation were set to 0.7, and 1, and we limited the maximum number of newly generated tokens to 100. Moreover, we employed the Adam (Kingma and Ba, 2014) optimizer, and the learning rates for Vicuna-1.5 and LLaVA-1.5 were set to 2e-4, and 2e-5 respectively with cosine scheduling. For LRMs (BERT and RoBERTa) and Multimodal LRMs (ConcatRoBERTa and VisualBERT), we used a maximum length of 512 tokens, with the learning rate set to 1e-5 with Adam optimizer. We trained them for 20 epochs and chose the best model evaluated on the dev set for evaluation of the test datasets.

Similarly, in all cases of the Appendix C, the temperature and number of beams for text generation were set to 0.7, and 1, and we limited the maximum number of newly generated tokens to 100. Also, we utilized the Adam optimizer, and the learning rates for BLIP-2 and LLaVA-1.5 were 5e-4, and 2e-4 respectively with cosine scheduling.

We employed the Parameter-Efficient Fine-Tuning (PEFT) (Mangrulkar et al., 2022) and Low-Rank Adaptation of Large Language Models (LoRA) (Hu et al., 2021) techniques for fine-tuning of large models, i.e., Vicuna, LLaVA, and BLIP-2.

B In-Context Learning: Results & Discussion

In this section, the results of zero- and few-shot experiments are illustrated in Table B.1. One thing worth mentioning about this section is that for the LLaVA-1.5 few-shot experiments, for the examples (shots), we only had the text written on the memes and the captions (with no images). This was due to a limitation in the implementation of LLaVA-1.5 that only accepted one image as the input. We defer exploration of the examples with more than one image for in-context learning of the LLaVA-1.5 model to future work.

C Meme Captioning Results

To generate captions for memes, first, we compared two state-of-the-art models, namely BLIP-2 and LLaVA-1.5-7B. We fine-tuned the Q-Former part of BLIP-2 for meme captioning. The vision encoder (CLIP-ViT (Radford et al., 2021)) and the LLM (OPT-6.7B (Zhang et al., 2022)) components of BLIP-2 are frozen by design. Regarding fine-tuning LLaVA, we have a few variations. First, we only fine-tuned the projector MLP that bridges between two modalities. As the second approach, we fine-tuned both the projector and the LLM (i.e., Vicuna-1.5-7B) together. In both variations, the vision encoder is frozen.

We fine-tuned each model for 1 epoch on the MemeCap dataset. Our results show the superiority of LLaVA-1.5-7B over BLIP-2, therefore, we chose to use fine-tuned LLaVA-1.5-7B for the meme captioning. To further optimize our pipeline, we tried another variation. We tested the case where in addition to the meme caption included in the MemeCap dataset, what would happen if we also used Optical Character Recognition (OCR), utilizing EasyOCR[3],

[3] https://github.com/JaidedAI/EasyOCR

Model	Shot(s)	H-F1	H-Precision	H-Recall
Vicuna-1.5 (text)	0	15.37	31.13	10.21
Vicuna-1.5 (text+caption from LLaVA-1.5)	0	17.60	30.28	12.40
LLaVA-1.5 (image)	0	17.74	27.10	13.18
LLaVA-1.5 (image+text)	0	20.39	30.27	15.38
LLaVA-1.5 (image+text+caption from LLaVA-1.5)	0	19.30	25.91	15.38
Vicuna-1.5 (text)	3	38.26	**34.73**	42.58
Vicuna-1.5 (text+caption from LLaVA-1.5)	3	35.89	33.98	38.02
LLaVA-1.5 (image+text)	3	24.78	27.87	22.31
Vicuna-1.5 (text)	5	**40.70**	33.39	**52.11**
Vicuna-1.5 (text+caption from LLaVA-1.5)	5	36.5	31.97	42.51
LLaVA-1.5 (image+text)	5	25.80	27.86	24.03

Table B.1: Comparison of results proposed methods in an in-context learning (zero- and few-shot learning). H-F1, H-Precision, and H-Recall are hierarchical F1, hierarchical precision, and hierarchical recall respectively. In LLaVA-1.5 few-shot experiments, due to the implementation limitation allowing only one image input, examples consisted solely of text from memes and their captions, lacking images. With an increase in the number of in-context examples, it appears that the model tends to perform better. However, due to LLaVA's restriction to only one image, the improvement is marginal compared to the enhancement achieved with text alone.

as illustrated in Figure 1, to extract the text written on the meme and feed that to the model as well since in the Persuasion dataset we have this data for each meme. We also tried both BLIP-2 and LLaVA in a zero-shot setting to assess their ability for image captioning without fine-tuning as well.

As discussed in Section 4, we used MemeCap dataset to fine-tune MLLMs for meme caption generations. Table C.1 shows the performance of the various models. From these results, initially, we chose to use LLaVA-1.5-7B with both the projector and LLM fine-tuned with OCR data for caption generation, as it outperformed other methods. However, as discussed earlier, we observed that even the caption generated by LLaVA-1.5-7B had some issues potentially leading to degraded performance on the Persuasion dataset. Therefore, we chose to create captions utilizing GPT-4 in a zero-shot configuration for our final results. In Section 4, the positive effect of this change is discussed in more detail with empirical evidence.

To compare different models for caption generation, we used Bertscore (Zhang et al., 2019) (using *microsoft/deberta-xlarge-mnli* model (He et al., 2021)), BLEU score (Post, 2018), and ROUGE-L (Lin, 2004) as evaluation metrics for the quality of generated captions. Bertscore assesses semantic similarities between the generated captions and the corresponding references using cosine similarity. In contrast, ROUGE-L and BLEU score rely on evaluating n-gram overlap between the generated captions and reference captions.

D Prompts for Caption Generation with GPT-4

As mentioned in Section 3, in addition to LLaVA-1.5, we used GPT-4 to generate captions for memes. LLaVA-1.5 provided a strong foundation for understanding the content and sentiment of the memes, while GPT-4's creative text generation capabilities helped us generate more informative captions. This allowed us to explore the potential of GPT-4 for generating captions that are not only relevant to the meme content but also capture the humor and cultural references often associated with memes. However, because of some of the meme's contents, it sometimes prevented generating captions to not generate toxic information. Table D.1 illustrates our prompts for obtaining captions using GPT-4. Given the sensitivity of GPT-4 to the content of this dataset, if the first prompt failed, we utilized the second prompt. In instances where there was another failure—constituting less than 10 samples in every 1000 examples—we employed our fine-tuned LLaVA model to generate captions for those samples.

E Hierarchical Evaluation Metrics

Hierarchical classification involves organizing classes in a hierarchy, where each class has a parent

Model	F1-Bertscore	ROUGE-L	BLEU-4
BLIP-2 (fine-tuned)	58.00	26.39	47.93
LLaVA-1.5 (projector fine-tuned)	59.01	27.41	**57.78**
LLaVA-1.5 (LLM & projector fine-tuned)	59.23	27.40	45.53
LLaVA-1.5 (projector fine-tuned + OCR data)	<u>59.80</u>	**28.08**	53.33
LLaVA-1.5 (LLM & projector fine-tuned + OCR data)	**59.90**	<u>27.86</u>	<u>53.86</u>
BLIP-2 (zero-shot)	50.30	12.88	31.81
LLaVA-1.5 (zero-shot)	55.11	19.31	40.15

Table C.1: Performance comparison of meme captioning models on MemeCap test set. In this table "+ *OCR data*" means for the training data we also appended the extracted text from the meme to help with the task of captioning the memes. The fine-tuned versions of the models yield superior captions, with all LLaVA iterations outperforming BLIP. The most effective model is LLaVA when both the language model and projector are tuned, particularly when incorporating text within the image generated by the OCR model.

Prompt
`Memes are one of the most popular types of content used in an online disinformation campaign. They are mostly effective on social media platforms since there they can easily reach a large number of users. This is a meme with the following text written inside the meme: "{meme_text}". In no more than 200 words, write a caption for this meme and say what is the meme poster trying to convey?`
`Memes are one of the most popular types of content used in an online disinformation campaign. They are mostly effective on social media platforms since there they can easily reach a large number of users. Memes in a disinformation campaign achieve their goal of influencing the users through a number of rhetorical and psychological techniques, such as causal oversimplification, name calling, smear. Identifying these memes are very useful and it can help to remove them from the internet and have a better and more calm place. To do so I want your help. I want to create a caption and find what this meme is trying to convey in order to train a model to find these memes. I provided a meme to you. In no more than 200 words, write a caption for this meme and say what is the meme poster trying to convey?`

Table D.1: These prompts were utilized to generate captions using GPT-4. Due to the sensitivity of GPT-4 to this dataset, if the first prompt failed to produce satisfactory results, we resorted to the second prompt.

or child relationship with other classes. In hierarchical classification tasks, Kiritchenko et al. (2006) introduced several key definitions to form a foundation for evaluating performance metrics which will be discussed in this section.

E.1 Partial Ordering and Hierarchy

A *partially ordered set (poset)* is denoted as $H = \langle C, \leq \rangle$, where C is a finite set and $\leq \subseteq C \times C$ is a reflexive, anti-symmetric, transitive binary relation on C. The hierarchy is defined by parent-child relationships between categories.

E.2 Hierarchical Categorization Task

A *hierarchical categorization task* involves assigning a boolean value to pairs $\langle d_j, c_i \rangle \in D \times C$, where D is a domain of instances, and $C = \{c_1, \ldots, c_{|C|}\}$ is a set of predefined categories with a given poset structure $H = \langle C, \leq \rangle$.

E.3 Hierarchical Consistency

A label set $C_i \subseteq C$ assigned to an instance $d_i \in D$ is considered *consistent* with a given hierarchy if C_i includes complete ancestor sets for every label $c_k \in C_i$. Hierarchical consistency ensures that assigned labels indicate the instance's position in the category hierarchy.

E.4 Hierarchical Precision, Recall, and F1 Score

For hierarchical evaluation, we introduce *hierarchical precision (HP)* and *hierarchical recall (HR)*. Each example belongs not only to its class but also to all ancestors of the class, except the root. The combined *hierarchical F1 score* is calculated using precision and recall with equal weights. Here are the formulas:

$$HP = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i|}{\sum_i |\hat{C}'_i|}$$

$$HR = \frac{\sum_i |\hat{C}_i \cap \hat{C}'_i|}{\sum_i |\hat{C}_i|}$$

$$Hierarchical\ F_\beta = \frac{(\beta^2 + 1) \cdot HP \cdot HR}{\beta^2 \cdot HP + HR}$$

Here, $\hat{C}_i$ and $\hat{C}'_i$ represent the extended sets of real and predicted classes, respectively, including their ancestor labels. Also $\beta \in [0, +\infty)$ and by using $\beta = 1$ we will have hierarchical F1. In the context of hierarchical classification, data is organized into a hierarchy of classes or categories, with each class having a parent-child relationship. The Hierarchical F1 score takes into account both precision and recall at different levels of the hierarchy, providing a comprehensive measure of a model's ability to correctly classify instances at various levels while considering the hierarchical structure of the classes. It balances the trade-off between false positives and false negatives within the hierarchy, offering a more nuanced assessment of classification performance in hierarchical data structures.

These hierarchical metrics provide a comprehensive evaluation of classification performance in the context of hierarchical categorization tasks.

Pauk at SemEval-2024 Task 4: A Neuro-Symbolic Method for Consistent Classification of Propaganda Techniques in Memes

Matt Pauk
University of Colorado Boulder
matt.pauk@colorado.edu

Maria Leonor Pacheco
University of Colorado Boulder
maria.pacheco@colorado.edu

Abstract

Memes play a key role in most modern information campaigns, particularly propaganda campaigns. Identifying the persuasive techniques present in memes is an important step in developing systems to recognize and curtail propaganda. This work presents a framework to identify the persuasive techniques present in memes for the SemEval 2024 Task 4, according to a hierarchical taxonomy of propaganda techniques. The framework involves a knowledge distillation method, where the base model is a combination of DeBERTa and ResNET used to classify the text and image, and the teacher model consists of a group of weakly enforced logic rules that promote the hierarchy of persuasion techniques. The addition of the logic rule layer for knowledge distillation shows improvement in respecting the hierarchy of the taxonomy with a slight boost in performance.

1 Introduction

Propaganda has long been used in media as a communication technique to influence people to subscribe to a particular idea or ideology. Identifying the presence of propaganda in media is an important subtask in building systems that can curtail the effect of propaganda campaigns. In modern times, a common way for propaganda to be spread is via memes. A meme is a either a short video or image, often overlayed with text, that is widely circulated over the internet. Identifying the presence and type of persuasion techniques used in memes is an important problem to be solved. The organizers of SemEval 2024 Task 4 propose a shared task for this very problem (Dimitrov et al., 2024). The shared task involves three sub-tasks: sub-task 1, identifying the persuasion technique(s) involved in the textual content of the meme; sub-task 2a, identifying which persuasion techniques are involved in both the visual and textual content of the meme; and sub-task 2b, identifying whether or not any persuasion technique is present in the visual and textual content of the meme. While the training data for all sub-tasks is in English, the evaluation phase includes different test sets for English, Arabic, Bulgarian, and North Macedonian. The proposed framework will focus only on the English version of the first two sub-tasks. Both 1 and 2a are hierarchical multi-label classification problems, where the goal is to classify the correct persuasion techniques used in the meme. The hierarchical nature of the persuasion techniques adds an extra element to this classification. All of the possible persuasion techniques are organized in a Directed Acyclic Graph (DAG), and full credit for a prediction is only given when the correct leaf node is predicted. Partial credit is given when any of its ancestors are given as a prediction.

There is a lot of previous work in the area of propaganda and persuasion identification. Much of this work has been based on previous SemEval shared tasks. The SemEval 2021 Task 6 is almost identical to the task explored here, without the hierarchical label structure (Dimitrov et al., 2021). The best-performing approaches on this task consisted of the use of a fine-tuned, text-based transformer for the textual content of the meme, some CNN or transformer based vision model to extract features from the image, and then a consolidation of the resulting embeddings via simple aggregation such as concatenation or average (Tian et al., 2021; Feng et al., 2021). Another SemEval task in 2023 focused on the identification of propaganda techniques in the text of news articles (Piskorski et al., 2023). In this case the best models used fine-tuned BERT based transformers (Wu et al., 2023; Hromadka et al., 2023).

The approach presented in this paper will also leverage a combination of a text-based transformer and a visual neural model. The key addition will be the incorporation of logic rules that encode the relationship between the possible hierarchical output classes. These relationships can be modeled by

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1424–1434

simple rules where the presence of a particular persuasion technique implies that its parent technique in the hierarchy is also present. For example, the rule `Straw Man` $\implies$ `Distraction` suggests that examples with the *Straw Man* persuasive technique also have it's parent technique *Distraction*. These rules will be used to promote predictions of persuasive techniques that respect the hierarchy. To test hierarchical consistency, we measure the number of hierarchy violations in predictions, where the model predicts a persuasive technique but not one of its ancestors in the hierarchy.

To incorporate logic rules, we take inspiration from the teacher-student logic rule framework proposed by Hu et al. (2016) that distills the information encoded in logic rules into the neural network parameters. The focus of this work is to explore if the incorporation of logic rules can improve on the results of neural based modes, while also producing more consistent results. The intuition is that distilling these hierarchical relationships into the weights of the network will allow the network to better recognize patterns that correspond to types of persuasive techniques, and ultimately make better predictions. We show that the addition of these logic rules does result in much more consistent predictions with a slight improvement in F1 scores.

2 Background and Related Work

Propaganda and Persuasion There is ample existing work in the identification of propaganda techniques. A similar SemEval task was proposed in 2021 to classify memes without including the hierarchy requirement (Dimitrov et al., 2021). Feng et al. (2021) proposed a framework for this task involving a text-based transformer built on RoBERTa, a visual feature extractor, and then a final transformer which takes as input the output of RoBERTa and the visual feature extractor. The authors consider two methods for this final encoder, a text pretrained transformer and a multi-modal transformer. The multi-modal transformer approach works the best, and they slightly improve on this score by combining several models together in an ensemble, which gives them state of the art results for the task. Tian et al. (2021) take a slightly different approach with this task. They similarly decompose the problem into a BERT based transformer and a visual feature extractor, but use a simpler method for combining the output embeddings by simply concatenating the output features before using a final classifier. They also experiment with a few different types of image feature extractors: a model specifically tuned to recognize faces, an Optical Character Recognition (OCR) model tuned to recognize text in an image, and the best performing extractor, which is a region based image feature extraction model that feeds into a multi-modal model.

Other previous SemEval tasks have focused just on the identification of propaganda techniques in text by leveraging multilingual datasets of news articles (Piskorski et al., 2023). Wu et al. (2023) and Hromadka et al. (2023) presented the two top performing systems for this task. Both use very similar approaches, leveraging a fine tuned version of RoBERTa for classification. Wu et al. (2023) had an interesting additional class weighting mechanism to try to improve performance on the under represented classes in the dataset. The approach proposed will leverage a lot of the same ideas regarding the usage of BERT based transformers for textual content, as well as visual feature extractors. Where this approach differs is in the incorporation of logic rules representing the relationship between propaganda techniques.

Hierarchical Classification Hierarchical multi label classification problems are split into two types: local methods, where an independent classifier is used for each node or for each level of the hierarchy; and global methods, which consider the entire hierarchy all at once (Levatić et al., 2014). In this paper, the interest is in exploring a global method that leverages logic rules to represent the relationship between classes in the hierarchy. Similar work has already been done in this area. Giunchiglia and Lukasiewicz (2021) propose a Coherent Hierarchical Multi-Label Classification Network (C-HMCNN) which uses a constraint layer on top of the regular network, as well as a specialized loss function to require the hierarchical constraints be satisfied. The constraint used is a simple one: the output probability of a subclass of a particular class in the hierarchy must be less than or equal to the output probability of its super-class. The approach for this project will be similar, but will follow more closely to the general logic rule integration method proposed by Hu et al. (2016). This framework consists of a teacher-student network, where the teacher network encodes logic rules as soft logic and distills that knowledge into the student network. This framework allows for more flexibility in the kinds of logic rules inte-

grated into a network. Additionally, it is less strict when enforcing the rules on the outputs of the neural network, allowing for a better balance between the signal coming from the direct supervision and the hierarchical knowledge.

3 System Overview

The proposed model architecture is based on a student-teacher knowledge distillation approach consisting of several components, which vary depending on the sub-task, but share a common structure. Regardless of task, base classifiers are used to encode raw textual and/or visual content. These resulting embeddings are then concatenated and used for predictions by the student network. The teacher network consists of a logic rule layer on top of the base student model that encodes the hierarchical information of the propaganda techniques. This logic layer is based on the teacher-student framework proposed by Hu et al. (2016). The framework distills the knowledge from the teacher network into the student network by training the student network to simultaneously emulate the gold labels and the teacher predictions.

3.1 Base Classifiers

Textual Model The textual model used will be the transformer-based model DeBERTa. DeBERTa was chosen based on its state of the art performance on short text datasets (Karl and Scherp, 2023). Additionally, this model was shown to do well on previous approaches for a very similar SemEval task (Feng et al., 2021; Tian et al., 2021). The version used is pre-trained version of DeBERTa proposed by He et al. (2021). This model is then fine-tuned on the textual meme content supplied by the SemEval 2024 task 6 organizers (Dimitrov et al., 2024).

Visual Model A ResNet-50 architecture is used for the vision component of sub-task 2a (He et al., 2016). This CNN based model is a medium sized model that achieved impressive results on the ImageNet classification task (Deng et al., 2009).

Combining the Textual and Visual Models The outputs of both textual and visual models need to be considered when making a prediction and therefore need to be combined in some way. A simple concatenation will be used and then inputted into a final feed forward network with sigmoid activations for each label.

3.2 Hierarchical Constraints

To introduce hierarchical constraints, we use a logical constraint layer on top of the base classifiers. The role of this layer is to distill knowledge coming from a set of logic rules into the weights of the classifiers. The implementation of this layer is based on the framework originally proposed by Hu et al. (2016) for the tasks of sentiment analysis and named entity recognition. The logic layer takes as input the sigmoid output for each label type from the base model and applies softened logic rules as regularization terms to obtain new predictions. The base network then learns weights based not only on the gold labels, but also based on the outputs of this logic rule regularized layer. This joint learning task is described by the following weight update equation which is a slightly modified version of the original formulation by Hu et al. (2016).

$$\theta^{t+1} = arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} l_1(y_n, \sigma_\theta(x_n)) + l_2(s_n^t, \sigma_\theta(x_n)) \quad (1)$$

where θ represents the network parameters, N is the number of samples, l_1 is the loss function for the student network and gold labels, l_2 is the loss function for the student network predictions and teacher network predictions, y_n are the gold labels, $\sigma_\theta(x_n)$ is a vector of label probabilities outputted by the base network, and s_n^t is the output of the logic rule layer.

The output of the logic-rule based layer s_n^t is obtained by evaluating the following equation, also originally formulated by Hu et al. (2016).

$$q^*(Y|X) = p_\theta(Y|X) \exp\left\{ -\sum_{l,g_l} \lambda_l (1 - r_{l,g_l}(X,Y)) \right\} \quad (2)$$

Where $p_\theta(Y|X)$ is the output of the base model, λ is a weighting parameter used to determine how strictly to follow a particular rule, and r_{l,g_l} is a softened first order logic rule. The strategy for softening as well as the rules used for this particular application are described in the next section.

Representing Logic Rules This framework supports any FOL rules that can be grounded in the inputs, the output probabilities of the base network, and/or the gold labels. The rules can be softened using the following t-norms as found in the work done by Bach et al. (2017) regarding probabilistic

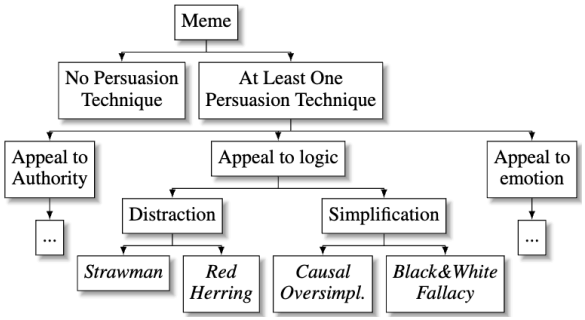


Figure 1: A subset of the hierarchy of propaganda techniques, provided by Dimitrov et al. (2024)

soft logic.

$$A \wedge B = \max(A + B - 1, 0)$$
$$A \vee B = \min(A + B, 1)$$
$$\neg A = 1 - A$$

After the rules have been converted to continuous representations they can be incorporated into the network via Equation (2). The following section describes the constraints that will be used for this specific classification task.

Hierarchical Logic Rules The rules for this application will focus on the hierarchical relationship between propaganda technique labels. The hierarchy of these labels is provided by the SemEval 2024 task 4 organizers, a sub-section of the hierarchy is shown in Figure 1 (Dimitrov et al., 2024). The full hierarchy includes 22 different persuasion techniques.

Given that the dataset is not balanced between all possible labels, we hypothesize that incorporating some of this hierarchical information into the model should allow for better prediction on lower coverage labels based on their relationship in the hierarchy to higher coverage labels. This hierarchical information will be encoded via logic rules that can be represented in the constraint layer via Equation (2), and will be distilled into the base model parameters via Equation (1) to help improve predictions.

The entire hierarchy can be represented via a sequence of rules:

$$\forall l \in L \text{ s.t. } l \neq root,\ l \implies par(l) \tag{3}$$

where L is the set of all possible labels and $par(l)$ represents the parent node of l in the hierarchy. Restructuring this rule slightly and grounding it in the outputs of the base model gives the following result:

$$\forall x \in X, \forall l \in L \text{ s.t. } l \neq root,\ \neg\sigma_l(x) \vee \sigma_{par(l)}(x) \tag{4}$$

where $\sigma_l(x)$ represents the probability that example x contains label l. Softening this logic gives the following expression for each rule:

$$\min(1 - \sigma_l(x) + \sigma_{par(l)}(x), 1) \tag{5}$$

Intuitively, these rules will enforce the hierarchy by penalizing predictions where the probability of a particular label is high, but the probability of its parent label is low.

4 Experimental Setup and Results

The method outlined above is evaluated on two separate but related tasks provided by the organizers of the SemEval 2024 Task 4 (Dimitrov et al., 2024). The details of these tasks and results are described below. Model implementations for both sub-tasks leverage Tensorflow for modeling and Hugging Face for the DeBERTa and ResNet models (Abadi et al., 2015; Wolf et al., 2019). [1]

4.1 SubTask 1

Experimental Setup The goal of subtask 1 is to identify which of the 20 persuasion techniques are present in the textual content of a meme. The dataset contains 7,000 labeled examples in the train set, 500 examples in the validation set, 1,000 in the dev set, and 1,500 in the test set. Hyper-parameter tuning is done on the validation set, with the final evaluation done on the dev and test sets.

Hierarchical F1, precision, and recall are used as the main evaluation metrics as defined by the SemEval Task organizers. However, micro F1, macro F1, and a count of the number of hierarchical violations are used as supplementary evaluation metrics. The hierarchical F1 score was originally formulated by Kiritchenko et al. (2006). This metric is the micro F1 of the label predictions, including both the actual persuasive techniques and their ancestor node categories in the hierarchy. The hierarchical violation metric is a count of the number of final true predictions which have an ancestor incorrectly marked as false. This metric is specifically used to evaluate whether the hierarchical logic rules are having an effect. The models presented will be compared against other approaches on the same

[1] Code can be found here: https://github.com/mappauk/Neuro-Symbolic-Final-Project

Rank	Model	HF1	Precision	Recall
1	914isthebest	0.752	0.684	0.836
2	BCAmirs	0.699	0.668	0.732
3	OtterlyObs...	0.697	0.648	0.755
⋮	⋮	⋮	⋮	⋮
15	**Pauk**	**0.627**	**0.716**	**0.573**
⋮	⋮	⋮	⋮	⋮
31	Baseline	0.369	0.477	0.300
32	WhatsaMeme	0.347	0.347	0.346
33	IIMAS1UTM1...	0.199	0.755	0.115

(a) Sub-task 1 (English) test set evaluation leaderboard. Our system, Pauk, places 15th.

Rank	Model	HF1	Precision	Recall
1	HierarchyEv...	0.746	0.867	0.655
2	NLPNCHU	0.707	0.782	0.645
3	BCAmirs	0.705	0.784	0.641
⋮	⋮	⋮	⋮	⋮
7	**Pauk**	**0.675**	**0.745**	**0.617**
⋮	⋮	⋮	⋮	⋮
12	BDA	0.504	0.477	0.493
13	Baseline	0.447	0.688	0.331
14	WhatsaMeme	0.366	0.313	0.440

(b) Sub-task 2a (English) test set evaluation leaderboard. Our system, Pauk, places 7th.

Table 1: Subtask 1 and 2a (English) test set evaluation results.

Model	Micro F1	Macro F1	Violations
NeuroSym(10)	0.574 ±0.009	0.283 ±0.015	13 ±7.8
NeuroSym(100)	0.583 ±0.006	0.301 ±0.018	7 ±4.2
NeuroSym(Max)	0.558 ±0.009	0.263 ±0.008	5 ±0.8
Baseline	0.581 ±0.002	0.307 ±0.018	42 ±20.9

Table 2: Sub-task 1 validation set results, comparing versions of NeuroSym with varying rule confidences against a baseline model leveraging only the classifiers and no logic layer.

Model	Micro F1	Macro F1	Violations
NeuroSym(10)	0.654 ±0.001	0.329 ±0.001	14.3 ±6.3
NeuroSym(100)	0.654 ±0.006	0.316 ±0.002	14.3 ±11.1
NeuroSym(Max)	0.651 ±0.003	0.329 ±0.01	13.3 ±5.8
Baseline	0.650 ±0.004	0.330 ±0.007	30.6 ±1.9

Table 3: Subtask 2a validation set results, comparing versions of NeuroSym with varying rule confidences against a baseline model leveraging only the classifiers and no logic layer.

task, as well as against themselves to measure the impact of the logical constraints.

The final hyper-parameters selected after tuning and used for evaluation on the dev/test sets are a learning rate of 3e-5, batch size of 4, and dropout of 0.1 after the BERT layer. Additionally, the teacher network rule confidences, represented by λ in Equation (2), are set to 100 for all rules. For dev set and validation set evaluations, the model is trained for 2 epochs on the training set. The model used to for the test set evaluation is trained for 3 epochs on the combined train and dev sets. Binary cross entropy is used as the loss function between the gold labels and student predictions, while KL Divergence is used between the student and teacher predictions.

Results The evaluation results for sub-task 1 on the English test set are shown in Table 1a. Dev set results can be found in Appendix A.1. Our system ranks 15th out of 33 submissions against the test set, as evaluated on the metric of hierarchical F1. The hierarchical F1 metric is a measure of micro F1 over all possible classes in the hierarchy after performing a post hoc operation to add the ancestors of predicted techniques to the list of predicted techniques for a particular example. Due to our systems focus on consistency of predictions with respect to the hierarchy, we also evaluate our system on micro F1 over all possible techniques without this post-hoc operation. Table 2 compares the average results of 3 runs against the validation set for our neuro-symbolic model with varying rule confidences of 10, 100, and Python's `sys.maxint`. In addition, a baseline version of the model without the hierarchical logic rule layer on top is added for comparison. The results for the individual runs can be found in Appendix A.2. Along with the F1 metrics, we present a measure of hierarchical violations over all predictions made.

The results show that regardless of the rule confidence used, the logic rule layer makes a noticeable improvement with regard to violations in the hierarchy of outputted predictions, with stronger rule confidences leading to less violations. This suggests that the rules are having their intended effect of making predictions consistent with the hierarchy. The results also show that very strong rule confidences seem to have a negative effect on F1 scores without much improvement in violations. The rule confidence of 100 seems to have the best compromise between consistent predictions and F1 scores, with even a slight improvement in F1 over the baseline model.

4.2 Sub-Task 2a

Experimental Setup The goal of sub-task 2a is to identify which of the 22 persuasion techniques are present in the textual and image content of a

Model	Micro F1	Macro F1
NeuroSym	0.374 ±0.014	0.162 ±0.006
Baseline(H)	0.433 ±0.016	0.227 ±0.017
Baseline	0.429 ±0.002	0.167 ±0.005

(a) Subtask 1 leaf node evaluation against the validation set.

Model	Micro F1	Macro F1
NeuroSym	0.474 ±0.005	0.218 ±0.005
Baseline(H)	0.488 ±0.005	0.233 ±0.02
Baseline	0.498 ±0.005	0.245 ±0.01

(b) Subtask 2a leaf node evaluation against the validation set.

Table 4: Leaf node evaluation to measure the effectiveness of the hierarchy usage in performance on leaf node propaganda technique predictions.

meme. The dataset distribution into train, validation, dev, and test is the same as task 1. The same hyper-parameter tuning method, final selected hyper-parameters, and evaluation metrics as used in sub-task 1 are also used here.

Results The results for sub-task 2a on the test set are displayed in Table 1b, with dev set results in Appendix A.1. Similar to sub-task 1, our model is in the middle of the pack, ranking 7th out of 14 for submissions on the test set. Table 3 is similar to Table 2 for sub-task 1, showing the results of experimenting on the validation set with varying rule confidences and a comparison to a baseline with no logic rule layer. Once again, we observe that the logic layer is leading to more hierarchically consistent predictions and a slight improvement in F1 scores. Additional results showing the model performance for each persuasive technique can be found in Appendix A.4.

Outside of consistency, one of the goals of using this logic rule student-teacher framework is to get the teacher model to distill information about the hierarchical relationship between the persuasive techniques into the student model and improve predictions on the actual leaf nodes representing specific persuasive techniques. In order to evaluate if this is actually the case, we perform an experiment evaluating just the predictions on the leaf nodes. For this experiment, the baseline model is trained and evaluated on only the leaf nodes of the hierarchy; Baseline(H) is trained on the full hierarchical data, but evaluated only on the leaf nodes; and NeuroSym includes the logic rule layer taking advantage of the hierarchical training data but also evaluated only on the leaf nodes. The results of the experiment averaged over three runs are shown in tables 4a and 4b. As shown in the results, the NeuroSym model has the lowest F1 scores when evaluated on both sub-task 1 and 2a. This indicates that the consistency enforced by the logic rule layer is actually negatively affecting leaf node predictions. The Baseline(H) model outperforms the baseline on sub-task 1 but performs worse on sub-task 2a, leaving inconclusive results as to whether the hierarchical data is helpful in leaf node prediction. The results of the individual runs, can be found in Appendix A.3.

5 Conclusion and Future Work

The framework presented attempts to solve the task of identification of persuasive techniques in memes. The key innovation involved in this framework compared to previous work done in this space is the integration of a logic rule knowledge distillation layer that weakly applies rules encoding the hierarchy of persuasion techniques. This layer is applied on top of a base model using a transformer based DeBERTa model for the textual component and a ResNet for the image component. We find that the logic rule network has some positive effect, consistently resulting in fewer hierarchical violations and a slight improvement in micro F1 scores. However, these logic rules do not lead to better predictions on the leaf node techniques themselves.

There were some difficulties in integrating these logic rules. The way the framework is set up, violations of the rules result in low probabilities for predictions by the teacher model. The part of the loss function that involves the KL divergence between these student and teacher predictions can cause the network to learn one of two aspects to minimize this loss. The first option is to raise the prediction probability of the rule violating ancestor label in the student network which will result in no rule violation and therefore no addition to the loss. Alternatively, the student network predictions can be lowered even further which also minimizes the KL divergence. The goal is for the former result to be learned, but it seems that often the latter is learned especially when the rule confidences are very high. Further work can be done to explore alternative logic rule interactions or loss function formulations to ensure the latter is always learned by the network.

In addition to improvements in the hierarchical logic rule integrations themselves, more work can be done to improve the base model by exploring other image processing methods outside of using a basic ResNet. Additionally, more intelligent ways of combining the textual hidden states and image hidden states can be explored, such as the use of a basic attention mechanism. Finally, the exploration of additional logic rules that promote parts of the textual content of an example that may indicate a particular persuasion technique could be experimented with. This may be especially useful for persuasion techniques with low coverage in the dataset.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2017. Hinge-loss markov random fields and probabilistic soft logic.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Zhida Feng, Jiji Tang, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, and Li Chen. 2021. Alpha at SemEval-2021 task 6: Transformer based propaganda classification. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 99–104, Online. Association for Computational Linguistics.

Eleonora Giunchiglia and Thomas Lukasiewicz. 2021. Multi-label classification neural networks with hard logical constraints. *Journal of Artificial Intelligence Research*, 72:759–818.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Timo Hromadka, Timotej Smolen, Tomas Remis, Branislav Pecher, and Ivan Srba. 2023. KInITVeraAI at SemEval-2023 task 3: Simple yet powerful multilingual fine-tuning for persuasion techniques detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 629–637, Toronto, Canada. Association for Computational Linguistics.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.

Fabian Karl and Ansgar Scherp. 2023. Transformers are short text classifiers: A study of inductive short text classifiers on benchmarks and real-world datasets.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*, pages 395–406. Springer.

Jurica Levatić, Dragi Kocev, and Sašo Džeroski. 2014. The importance of the label hierarchy in hierarchical multi-label classification. *Journal of Intelligent Information Systems*, 45:247—-271.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Junfeng Tian, Min Gui, Chenliang Li, Ming Yan, and Wenming Xiao. 2021. MinD at SemEval-2021 task 6: Propaganda detection using transfer learning and multimodal fusion. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1082–1087, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Ben Wu, Olesya Razuvayevskaya, Freddy Heppell, João A. Leite, Carolina Scarton, Kalina Bontcheva, and Xingyi Song. 2023. SheffieldVeraAI at SemEval-2023 task 3: Mono and multilingual approaches for news genre, topic and persuasion technique classification. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1995–2008, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Dev Results

Tables 5a and 5b show the results of our model on the dev set for both subtask 1 and subtask 2a respectively. Our model performs in the middle of the pack for both subtasks, finishing 16th out of 33 for subtask 1 and 7th out of 11th on subtask 2a.

A.2 Validation Result Individual Runs

Tables 6 and 7 show the individual runs on the validation set measuring the effectiveness of the logic rule layer for both subtasks. The baseline model uses the classifiers with no logic rule layer, while the NeuroSym models use the classifiers and the logic rule layer with varying levels of confidence in the logic rules.

Model	Micro F1	Macro F1	Violations
NeuroSym(10)	0.571	0.270	7
NeuroSym(10)	0.586	0.304	8
NeuroSym(10)	0.565	0.275	24
NeuroSym(100)	0.576	0.318	13
NeuroSym(100)	0.582	0.276	4
NeuroSym(100)	0.591	0.308	4
NeuroSym(Max)	0.567	0.271	6
NeuroSym(Max)	0.558	0.267	4
NeuroSym(Max)	0.545	0.252	5
Baseline	0.583	0.285	39
Baseline	0.583	0.329	69
Baseline	0.578	0.308	18

Table 6: Individual runs on the validation set for subtask 1.

Model	Micro F1	Macro F1	Violations
NeuroSym(10)	0.654	0.328	12
NeuroSym(10)	0.653	0.331	23
NeuroSym(10)	0.656	0.329	8
NeuroSym(100)	0.653	0.314	5
NeuroSym(100)	0.652	0.312	8
NeuroSym(100)	0.658	0.322	30
NeuroSym(Max)	0.649	0.340	7
NeuroSym(Max)	0.653	0.331	12
NeuroSym(Max)	0.651	0.316	21
Baseline	0.648	0.336	28
Baseline	0.657	0.320	32
Baseline	0.647	0.334	32

Table 7: Individual runs on the validation set for subtask 2a.

A.3 Leaf Node Evaluation Individual Runs

Tables 8 and 9 show the results of the individual runs for the leaf node experiment for both subtask 1 and 2a. The Baseline model uses just the base classifiers trained and evaluated only on the leaf nodes of the hierarchy. The Baseline(H) model also uses only the base classifiers and is evaluated on only the leaf nodes of the hierarchy, but is trained on the full hierarchical data. Finally, the NeuroSym model is also evaluated on the leaf nodes but leverages the logic rule layer and the full hierarchical data.

Model	Micro F1	Macro F1
NeuroSym	0.376	0.161
NeuroSym	0.356	0.155
NeuroSym	0.389	0.169
Baseline(H)	0.431	0.203
Baseline(H)	0.414	0.242
Baseline(H)	0.453	0.237
Baseline	0.428	0.174
Baseline	0.427	0.163
Baseline	0.431	0.165

Table 8: Individual runs on the validation set for subtask 1 evaluating only the performance of predictions on the leaf node propaganda techniques.

Rank	Model	HF1	Precision	Recall
1	CLaC	0.881	0.808	0.967
2	OtterlyObs...	0.690	0.636	0.754
3	GreyBox	0.685	0.657	0.716
⋮	⋮	⋮	⋮	⋮
16	**Pauk**	**0.611**	**0.654**	**0.573**
⋮	⋮	⋮	⋮	⋮
31	nowhash	0.495	0.379	0.711
32	SINAI	0.430	0.315	0.677
33	Baseline	0.358	0.466	0.291

(a) Subtask 1 dev set evaluation results. Our system, Pauk, is ranked 16th out of 33.

Rank	Model	HF1	Precision	Recall
1	BCAmirs	0.699	0.770	0.640
2	NLPNCHU	0.697	0.767	0.639
3	SuteAlbastre	0.688	0.675	0.700
⋮	⋮	⋮	⋮	⋮
7	**Pauk**	**0.669**	**0.715**	**0.629**
⋮	⋮	⋮	⋮	⋮
9	Lomonoso...	0.648	0.774	0.557
10	hariswaqar	0.646	0.703	0.598
11	Baseline	0.446	0.685	0.331

(b) Subtask 1 and 2a dev set evaluation results. Out system, Pauk, is ranked 7th out of 11.

Table 5: Subtask 1 and 2a dev set leaderboards.

Model	Micro F1	Macro F1
NeuroSym	0.471	0.215
NeuroSym	0.481	0.224
NeuroSym	0.471	0.214
Baseline(H)	0.494	0.259
Baseline(H)	0.488	0.230
Baseline(H)	0.483	0.209
Baseline	0.503	0.259
Baseline	0.491	0.234
Baseline	0.499	0.243

Table 9: Individual runs on the validation set for subtask 2a evaluating only the performance of predictions on the leaf node propaganda techniques.

A.4 Results By Propaganda Technique

Tables 10 and 11 show the results of the dev set predictions on a per class basis. For both subtasks, we see the best performance for those classes higher up in the hierarchy due to the presence of the logic rules in the network as well as a larger number of training examples. Unsurpisingly, we see very poor performance for those techniques with very few training examples, ex: Obfuscation, Reductio ad hitlerum, Straw Man, and Red Herring. Unexpectedly, we see the F1 scores decrease for many of the leaf node propaganda techniques for subtask 2a despite having access to much more training data and getting overall higher F1 scores in aggregate. It appears this lift in F1 is due to the increase in F1 for the higher up nodes in the hierarchy such as Ethos, Pathos, and Ad Hominem as well as a drastic increase in a few leaf node techniques such as Smears and Loaded Language. Additionally, we see several more techniques with a F1 score of 0. This suggests that the images are helpful for classifying high level propaganda techniques and certain leaf techniques but actually confuse the model and lead to worse predictions than the textual model for many of the leaf node techniques.

Class	F1	Precision	Recall	Examples
Logos	0.76	0.76	0.76	545
Repetition	0.39	0.43	0.35	46
Obfuscation, Intentional vagueness, Confusion	0.00	0.00	0.00	8
Reasoning	0.56	0.53	0.60	278
Justification	0.70	0.73	0.65	343
Slogans	0.39	0.54	0.31	111
Bandwagon	0.22	1.00	0.22	16
Appeal to authority	0.85	0.83	0.87	136
Flag-waving	0.48	0.62	0.39	89
Appeal to fear/prejudice	0.19	0.40	0.12	66
Simplification	0.46	0.47	0.44	215
Causal Oversimplification	0.20	0.39	0.13	53
Black-and-white Fallacy/Dictatorship	0.38	0.39	0.37	98
Thought-terminating cliché	0.21	0.24	0.18	78
Distraction	0.30	0.38	0.25	72
Misrepresentation of Someone's Position (Straw Man)	0.00	0.00	0.00	10
Presenting Irrelevant Data (Red Herring)	0.00	0.00	0.00	10
Whataboutism	0.23	0.47	0.15	52
Ethos	0.80	0.79	0.81	610
Glittering generalities (Virtue)	0.41	0.47	0.37	71
Ad Hominem	0.71	0.71	0.70	506
Doubt	0.18	0.26	0.13	45
Name calling/Labeling	0.50	0.64	0.40	262
Smears	0.52	0.51	0.52	282
Reductio ad hitlerum	0.00	0.00	0.00	11
Pathos	0.65	0.68	0.62	427
Exaggeration/Minimisation	0.31	0.62	0.21	62
Loaded Language	0.55	0.63	0.48	303

Table 10: Results for each propaganda technique when evaluated against the submitted dev predictions for subtask 1.

Class	F1	Precision	Recall	Examples
Logos	0.77	0.79	0.75	583
Repetition	0.04	1.00	0.02	46
Obfuscation, Intentional vagueness, Confusion	0.00	0.00	0.00	12
Reasoning	0.54	0.56	0.53	284
Justification	0.69	0.77	0.62	379
Slogans	0.35	0.44	0.30	115
Bandwagon	0.00	0.00	0.00	18
Appeal to authority	0.85	0.81	0.90	143
Flag-waving	0.48	0.50	0.46	123
Appeal to fear/prejudice	0.00	0.00	0.00	78
Simplification	0.51	0.49	0.54	214
Causal Oversimplification	0.00	0.00	0.00	56
Black-and-white Fallacy/Dictatorship	0.37	0.34	0.41	103
Thought-terminating cliché	0.26	0.28	0.24	78
Distraction	0.16	0.53	0.10	83
Misrepresentation of Someone's Position (Straw Man)	0.00	0.00	0.00	11
Presenting Irrelevant Data (Red Herring)	0.00	0.00	0.00	10
Whataboutism	0.14	0.71	0.08	62
Ethos	0.91	0.89	0.94	847
Glittering generalities (Virtue)	0.39	0.38	0.39	92
Ad Hominem	0.81	0.79	0.84	660
Doubt	0.13	0.50	0.08	52
Name calling/Labeling	0.57	0.60	0.54	261
Smears	0.73	0.67	0.80	504
Reductio ad hitlerum	0.00	0.00	0.00	16
Pathos	0.73	0.75	0.70	635
Exaggeration/Minimisation	0.03	1.00	0.01	68
Loaded Language	0.64	0.70	0.58	306
Transfer	0.40	0.59	0.30	274
Appeal to (Strong) Emotions	0.03	0.33	0.02	56

Table 11: Results for each propaganda technique when evaluated against the submitted dev predictions for subtask 2a.

Response to the reviewer comments for SemEval 2024 Submission #215

Paper title:
Saama Technologies at SemEval-2024 Task 2: Three-module System for NLI4CT Enhanced by LLM-generated Intermediate Labels

Reviewer 1:
There is no specific comment to work out.

Reviewer 2:

- Comment 1: It would be advantageous for the authors to include a more comprehensive review of literature relevant to their system. Specifically, an exploration of the foundational principles behind their approach, comparisons with analogous systems, and their applications across various fields and tasks in preceding studies would enrich the paper's context and depth.

  Response: Following this comment, we added more detailed literature search in the section 2.

- Comment 2: Enhancing the manuscript with highlighted examples of the system in action could significantly improve its readability. Currently, the dense text can obscure key points, making it challenging for readers to identify the central contributions and insights.

  Response: Following this comment, we added working example case as the texts in italic fonts in Figure 1.

- Comment 3: The length of the appendix is notably extensive, which is unprecedented in my experience. A more concise appendix, focused on essential information, would be more accessible and less daunting for readers.

  Response: While we fully understand the concern of the reviewer on the readability of the appendix, we also worry that anyone trying to reproduce our work would be troubled if we exclude any example we used in our prompts. So we added more subsections and explanations in Appendix C to make it more readable.

AmazUtah_NLP at SemEval-2024 Task 9: A MultiChoice Question Answering System for Commonsense Defying Reasoning

Mina Ghashami*
Amazon Web Services
ghashami@amazon.com

Soumya Smruti Mishra*
Amazon Web Services
soumish@amazon.com

Abstract

The SemEval 2024 BRAINTEASER task represents a pioneering venture in Natural Language Processing (NLP) by focusing on lateral thinking, a dimension of cognitive reasoning that is often overlooked in traditional linguistic analyses. This challenge comprises of Sentence Puzzle and Word Puzzle subtasks and aims to test language models' capacity for divergent thinking.

In this paper, we present our approach to the BRAINTEASER task. We employ a holistic strategy by leveraging cutting-edge pre-trained models in multiple choice architecture, and diversify the training data with Sentence and Word Puzzle datasets. To gain further improvement, we fine-tuned the model with synthetic humor/jokes dataset and the RiddleSense dataset which helped augmenting the model's lateral thinking abilities. Empirical results show that our approach achieve 92.5% accuracy in Sentence Puzzle subtask and 80.2% accuracy in Word Puzzle subtask.

1 Introduction

The success of language models has inspired the Natural Language Processing community to attend to tasks that require implicit and complex reasoning. Human reasoning encompasses two types of reasoning: lateral and vertical thinking approaches. Lateral thinking demand out-of-the-box thinking. It is a form of creative reasoning that deviates from traditional, logical processes and has received little attention from NLP community. Vertical thinking on the other hand, relies on logical reasoning, and have been relatively popular in the past few years.

The BRAINTEASER dataset by (Jiang et al., 2023) stands as a crucial benchmark for evaluating question-answering systems. It particularly assesses these systems on their ability for lateral thinking — pushing them to transcend conventional commonsense reasoning towards more innovative and creative approaches to problem-solving. Additionally, as part of this effort to test language models' lateral thinking capabilities, the SemEval 2024 BRAINTEASER task (Jiang et al., 2024), offers a focused challenge derived from the broader dataset, further probing the creative reasoning abilities of these models. This task is crucial because it addresses a gap in Natural Language Processing (NLP) where most tasks focus on linear, logical (vertical) thinking, neglecting the complex, divergent aspects of human cognition represented by lateral thinking. By encompassing two subtasks — Sentence Puzzle and Word Puzzle — BRAINTEASER aims to test a model's ability to go beyond conventional commonsense associations, requiring an understanding of both standard meanings and the ability to reinterpret them in novel ways. This is vital for advancing the field of NLP, as it pushes the boundaries of what artificial intelligence can achieve in terms of mimicking the nuanced and creative aspects of human thought.

Our system adopts a multifaceted strategy for this challenge, centering on the use of advanced pre-trained models like BERT (Devlin et al., 2019) and DeBERTaV3 (He et al., 2023) through HuggingFace's (Wolf et al., 2020) AutoModelForMultipleChoice and AutoModelForSequenceClassification. This approach is enhanced by a diverse training regimen that mixes Sentence and Word Puzzle datasets, ensuring a broad exposure to different types of lateral thinking challenges. Additionally, the model is fine-tuned with a humor/jokes dataset generated by GPT-4 (OpenAI et al., 2024) and the RiddleSense (Lin et al., 2021) dataset, which introduces elements of creativity, unconventional thinking, and complex puzzle-solving. This comprehensive strategy aims to equip the model with enhanced lateral thinking abilities, crucial for tackling the creative and nuanced demands of the BRAINTEASER task in SemEval 2024.

*Both authors contributed equally to this work.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1436–1442

In our participation in the BRAINTEASER task, we discovered that our system, particularly when finetuned with AutoModelForMultipleChoice, outperformed the baseline instruction-tuned systems mentioned in the original paper. This approach demonstrated a significant advantage in handling multiple-choice tasks. However, we faced challenges with AutoModelForSequenceClassification, suggesting an area for improvement. The incorporation of additional synthetic data and open-source dataset like RiddleSense positively influenced our performance. Quantitatively, our system achieved a commendable 6th place in the Sentence Puzzle and 10th in the Word Puzzle, indicating stronger proficiency in sentence-based challenges and room for growth in word-based puzzles.

Our code and data will be available at https://github.com/soumyasmruti/semeval-2024-brainteaser after cleaning and de-anonimization.

2 Background

The task involves two types of brain teasers: Sentence Puzzle and Word Puzzle. In Sentence Puzzle, the input is a sentence-based question that defies commonsense, with multiple-choice answers. For instance, "A man shaves everyday, yet keeps his beard long." The choices include "He is a barber," "He wants to maintain his appearance," and so on. The Word Puzzle involves a word-based teaser, like "What part of London is in France?" with choices focusing on letters in the words (e.g., "The letter N"). The output in both cases is the selection of the correct choice that represents lateral thinking.

In order to counter the potential for Large Language Models (LLMs) memorizing solutions, BRAINTEASER (Jiang et al., 2023) incorporates two novel methods of puzzle generation: semantic and context reconstruction. These techniques generate variations of puzzles that preserve the core challenge of overturning conventional commonsense reasoning without altering the fundamental nature of the puzzles. This approach is aimed at enhancing the robustness of the puzzles against the memorization capabilities of LLMs, ensuring that the puzzles continue to effectively test the models' ability to engage in lateral thinking by challenging ingrained commonsense assumptions. This is to ensure the model is evaluating reasoning ability rather than memorization.

Systems are evaluated based on two accuracy metrics: Instance-based Accuracy, considering each question (original and adversarial) as a separate instance, and Group-based Accuracy, where a system must correctly solve all questions in a group (original and its adversarial versions) to score.

3 Related Work

We can broadly categorize the reasoning landscape of language models into two groups. The first, is 'commonsense reasoning', also known as 'vertical reasoning'. This refers to the ability to make deductions based on everyday knowledge. The second category is 'lateral reasoning'; i.e. a creative problem-solving approach that involves looking at situations from unconventional perspectives.

Researchers have explored various approaches to endow LLMs with commonsense reasoning abilities (Rae et al., 2021). One prominent approach is the use of knowledge graphs, which represent structured knowledge in the form of entities and their relationships (Ilievski et al., 2021). Authors in (Wang et al., 2021) proposed a method for incorporating commonsense knowledge from ConceptNet (Speer et al., 2018) into language models, leading to improved performance on commonsense reasoning tasks.

Another approach involves fine-tuning pretrained LLMs on commonsense reasoning datasets. Authors in this paper (Huang et al., 2019) introduced the COSMOS QA dataset, which consists of multiple-choice questions that require commonsense reasoning. They showed that fine-tuning pretrained LLMs on this dataset can significantly improve their commonsense reasoning capabilities.

Researchers have also investigated the use of prompting techniques to elicit commonsense reasoning from LLMs without explicit fine-tuning. (Zhou et al., 2022) proposed a method called "Conditional Prompt-Tuning" that enables LLMs to perform commonsense reasoning by conditioning on carefully designed prompts. In another work (Wei et al., 2022), chain-of-thought prompting showed how to unlock LLM's reasoning ability via effective prompting techniques.

There hasn't been extensive research on 'lateral thinking' of LLMs. Very recenlty, OlaGPT (Xie et al., 2023) proposed a cognitive architecture framework in which they summarize various methods of human reasoning into Chain-of-Thought (CoT) templates, to maximize the LLMs' reasoning effect.

Overall, while LLMs have shown flashes of non-linear, exploratory thinking on some benchmarks, lateral thinking as a holistic cognitive process remains an open challenge.

4 Methodology

In this section, we describe different methods and approaches we employed in solving the BrainTeaser puzzle.

4.1 Sequence Classification with BERT

In this approach, we enhanced the performance of a sequence classification model through the instruction fine-tuning process. We leveraged the powerful contextual embeddings provided by BERT (Devlin et al., 2019). Our methodology involved initializing the model with pre-trained BERT weights and employing the streamlined 'AutoModelForSequenceClassification' class from the Hugging Face Transformers library, which linearly projects the embedding from the language model encoder to each document into the class logits for that document. We instructed the model with selecting the most appropriate answer from a set of four choices provided alongside a given question. Despite the meticulous fine-tuning process our experimental results revealed sub-optimal performance.

4.2 MultipleChoice QA with BERT and DeBERTa

We leveraged the versatile 'AutoModelForMultipleChoice' architecture from Hugging Face's library, which integrates a pre-trained transformer model with a specialized classification head. This architecture was pivotal in adapting the model for our multiple-choice task, which involved combining both Word Puzzle and Sentence Puzzle datasets to diversify our training data.

To ensure optimal performance, we split our training data into separate training and validation sets. Throughout the training process, we utilized the validation set to fine-tune hyperparameters, ensuring the model's efficacy.

The AutoModelForMultipleChoice architecture comprises a pre-trained base transformer augmented with a classification head. This head, typically consisting of neural network components such as linear layers and activation functions, enables the model to make informed multiple-choice predictions.

Our model initialization involved embedding pre-trained DeBERTa representations, followed by further training on the designated training dataset. This approach facilitated the model's adaptation to our specific task requirements, ultimately enhancing its performance.

4.2.1 Augmenting with RiddleSence and Humor Data

Next, we decided to use two additional data sources to augment our training data. This was with the aim of expanding the diversity of our dataset, enriching it with a wide range of humor styles, scenarios, and perspectives. This augmentation not only increases the robustness and variety of our model but also enhances its adaptability to different contexts. We utilized the public Riddlesense dataset as well as creating humor style data by prompting GPT 4.

The Ridlesense dataset consists of Riddles which are a form of puzzle where a question, often presented in a cryptic or metaphorical manner, challenges the reader to find a clever or unexpected answer.

To create the humor style QA, we prompted GPT 4. Crafting jokes content often requires a touch of ingenuity, an out-of-the-box approach, and a healthy dose of lateral thinking, and GPT-4 allowed us to explore unconventional and amusing angles to questions and answers. It's like having a comedy writer who never runs out of fresh and unexpected punchlines. The details about how the dataset was generate is provided in Appendix A.

We then used the same AutoModelForMultipleChoice architecture and trained the model on augmented training data.

5 Experimental setup

5.1 Datasets Description

The task dataset and additional datasets used in our approaches are detailed in Table 1, with all datasets being in the English language. We did not perform any extra pre-processing on the original training or test data. To generate humor data, we used GPT-4 (OpenAI et al., 2024) using prompt engineering. Regarding the RiddleSense (Lin et al., 2021) dataset, which originally had five labels, we adapted it to a four-label format. This was achieved by reassigning questions with the fifth label as the correct answer to the fourth choice. Consequently, all fifth-choice answers across questions were remapped to their corresponding fourth choices, and all original fifth choices were discarded. Riddlesense and humor datasets, were

	Sentence Puzzle			Word Puzzle		
Dataset	**Train**	**Validation**	**Test**	**Train**	**Validation**	**Test**
Provided	405	102	120	316	80	96
Humor Data GPT4	211	-	-	211	-	-
Riddlesense	4531	-	-	-	-	-

Table 1: Dataset Statistics, '-' means the data was not used for the stage of the task.

selected for their similarity to the original training data, offering commonsense-defying puzzles. For details on the train-validation-test split, please refer to Table 1. We also experimented by adding SWAG (Zellers et al., 2018) and CODAH (Chen et al., 2019) datasets, but found that they reduced overall performance.

5.2 Implementation Details

The raw text was tokenized using a byte-level Byte-Pair Encoding (BPE) vocabulary with 50,257 merge rules, and inputs longer than 1024 tokens were truncated.

Our models were based on the BERT-base and DeBERTaV3 base architectures. The BERT model comprises 12 layers, 768-dimensional embeddings, and 12 attention heads, totaling 117M parameters. The DeBERTaV3 base model features 12 layers and a hidden size of 768, with 110M backbone parameters and a 128K token vocabulary introducing an additional 98M parameters in the embedding layer.

Both models were initialized with pre-trained weights in the AutoModelForMultipleChoice architecture. We conducted a random hyperparameter search, exploring batch sizes of [4, 16, 32] and learning rates of [5e-5, 1e-4, 2e-4]. The configurations yielding the highest validation accuracy were selected for each model size.

We utilized Amazon SageMaker for training, opting for the ml.p3.8xlarge instance for BERT-based approaches and the ml.p3.16xlarge instance for training our DeBERTaV3-based approaches. The training time for the BERT models with the original data was under 20 minutes, while the DeBERTa-based approaches were trained in under one hour. This efficient use of resources enabled us to achieve significant performance improvements with minimal cost and time.

6 Results

In Table 2, we demonstrate the performance of our model, where the provided numbers represent the accuracy for various groups. "Original," "Semantic," and "Context" denote the original question, its semantic reconstruction, and context reconstruction, respectively. These three categories are based on instance-based accuracy, where each question is treated as a separate instance. The score reports the accuracy for both the original question and its adversarial counterparts. "Orig. + Sem." represents group-based accuracy, where the original question and its semantic reconstruction are considered and calculated together. Similarly, "Orig. + Sem. + Con." includes the previous group along with the contextual reconstruction of the original question.

In the table, "AMSC" represents AutoModelForSequenceClassification, and "AMMC" represents AutoModelForMultipleChoice. The models used are bert-base-uncased and microsoft/deberta-v3-base. The notation "train-data-wp+sp" indicates that the training data for this approach includes both sentence puzzle and word puzzle training data provided by the organizers of the task. "Humor" represents the synthetic dataset generated by prompting GPT-4, and "RiddleSense" refers to the open-source RiddleSense dataset (Lin et al., 2021). The scores of human performance and the baseline system, as provided in the original paper (Jiang et al., 2023), are depicted in gray. Scores obtained by our system are shown in black, with the best performances for each task highlighted in bold.

6.1 Subtask A : Sentence Puzzle

Initially, we trained our models only on the provided sentence puzzle dataset but soon realized that combining both the sentence puzzle and word puzzle datasets yielded better validation scores. Consequently, we used the bert-base model with AutoModelForSequenceClassification, achieving an overall accuracy of 50.8%. Given that the dataset is in a multiple-choice format, we experimented

Approaches	Sentence Puzzle						Word Puzzle					
	Original	**Semantic**	**Context**	**Orig. + Sem.**	**Orig. + Sem. + Con.**	**Overall**	**Original**	**Semantic**	**Context**	**Orig. + Sem.**	**Orig. + Sem. + Con.**	**Overall**
Human	.907	.907	944	.907	.889	.920	.917	.917	.917	.917	.900	.917
ChatGPT	.608	.593	.679	.507	.397	.627	.561	.524	.518	.439	.292	.535
RoBERTa-L	.435	.402	.464	.330	.201	.434	.195	.195	.232	.146	.061	.207
BERT-base + AMSC + train-data-wp+sp	.475	.55	.5	.35	.25	.508	.281	.312	.375	.031	0	.323
BERT-base + AMMC + train-data-wp+sp	.650	.625	.625	.600	.500	.600	.438	.375	.406	.344	.375	.406
DeBERTaV3 + AMMC + train-data-wp+sp	.900	.900	.850	.900	.825	.883	.75	.75	.625	.719	.500	.708
DeBERTaV3 + AMMC + train-data-wp+sp + Humor + RiddleSense	**.925**	**.950**	**.900**	**.925**	**.875**	**.925**	-	-	-	-	-	-
DeBERTaV3 + AMMC + train-data-wp + Humor	-	-	-	-	-	-	**.844**	**.812**	**.750**	**.781**	**.594**	**.802**

Table 2: SemEval2024 Task 9: BRAINTEASER results table, which shows the performance of different approaches on the test set. Orig. = Original, Sem. = Semantic, Con. = Context, AMSC = AutoModelForSequenceClassification, AMMC = AutoModelForMultipleChoice

with AutoModelForMultipleChoice using the same bert model. This change significantly improved performance, increasing accuracy by 10 points to 60%. Encouraged by this, we opted for the larger DeBERTaV3 model under the AutoModelForMultipleChoice configuration. This model, combined with the original dataset, significantly boosted performance, raising overall accuracy to 83.3%. After incorporating additional datasets containing humor-style questions and the RiddleSense dataset, our best accuracy score reached 92.5%. Our approach ranked 6th among the 31 teams that participated in the task and outperformed the baseline zero shot ChatGPT by almost 50 percentage points.

6.2 Subtask B : Word Puzzle

The word puzzle setup followed almost the same approach as sentence puzzle but during validation process we found the best model was the one which was trained with only original training data from word puzzle dataset and adding humor dataset. Adding RiddleSense data and sentence pzzle data didn't improve the score of the word puzzle in validation process, therefore we didn't submit that output. Our approach for this subtask didn't perform that well when compared to other teams, we ranked 10th among the 23 teams that participated in this task, but outperformed the baseline zero shot ChatGPT by almost 40 percentage points.

7 Conclusion

In this work, we present our novel system designed for the SemEval 2024 BRAINTEASER task, which notably achieved 6th place in the Sentence Puzzle and 10th in the Word Puzzle categories. Our approach leverages advanced pre-trained models like BERT and DeBERTa, optimized through HuggingFace's AutoModelForMultipleChoice and AutoModelForSequenceClassification. This strategy was further enhanced by incorporating a diverse training regimen, blending Sentence and Word Puzzle datasets with a unique humor/jokes dataset and

the RiddleSense dataset. This mix has been instrumental in equipping our model with the lateral thinking capabilities essential for this task. While our system excelled in the Sentence Puzzle, reflecting a stronger grasp in sentence-based lateral thinking, the performance in the Word Puzzle highlighted areas for improvement, particularly in word-based lateral reasoning. The additional challenge posed by adversarial versions of puzzles, involving both Semantic and Context Reconstruction, underscores the complexity of this task. Our system's performance underscores the efficacy of our training approach in enhancing lateral thinking in language models, a significant step forward in NLP. Future work will focus on refining our approach for word-based puzzles and further enhancing the model's ability to navigate complex, creative reasoning paths, thereby advancing the field's understanding of AI's potential in mimicking nuanced aspects of human cognition.

References

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. Codah: An adversarially authored question-answer dataset for common sense. *arXiv preprint arXiv:1904.04365*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Filip Ilievski, Pedro Szekely, and Bin Zhang. 2021. Cskg: The commonsense knowledge graph. In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*, pages 680–696. Springer.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP 2021): Findings*. To appear.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak,

Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. Conceptnet 5.5: An open multilingual graph of general knowledge.

Bin Wang, Guangtao Wang, Jing Huang, Jiaxuan You, Jure Leskovec, and C-C Jay Kuo. 2021. Inductive learning on commonsense knowledge graph completion. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Yuanzhen Xie, Tao Xie, Mingxiong Lin, WenTao Wei, Chenglin Li, Beibei Kong, Lei Chen, Chengxiang Zhuo, Bo Hu, and Zang Li. 2023. Olagpt: Empowering llms with human-like problem-solving abilities. *arXiv preprint arXiv:2305.16334*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825.

A Humor Dataset Details

We used the following prompts to generate Jokes or Humor style dataset. We experimented with multiple prompts and gather all the output in a json file and analyzed them manually.

PROMPT 1 - Could you create a dataset for me that includes humor-styled questions, each with multiple choices and an answer? The dataset should be in JSON format.

PROMPT 2 - Could you create a dataset of 40 jokes for me in JSON format? Each joke should include four options and the correct answer.

PROMPT 3 - Could you generate an additional 20 jokes with multiple choices and an answer? Please ensure there are no duplicates and that none of them are the same as those previously generated.

We initially prompted GPT 4 to generate 200 questions at once but that didn't go well. The output contained duplicate questions after 15 / 16 unique ones. Basically, the model kept repeating itself. So we mostly used PROMPT 3 multiple times to generate high quality data. Before adding each we checked for duplicates again manually. Provided below are some of the jokes generated by the prompt.

> "joke": "Why did the bicycle fall over?", "options": ["A. Because it was two-tired.", "B. It had a flat.", "C. It was unbalanced.", "D. It slipped."], "answer": "A"
>
> "joke": "What's orange and sounds like a parrot?", "options": ["A. A carrot", "B. An orange bird", "C. A tangerine", "D. A flamingo"], "answer": "A" ,

IITK at SemEval-2024 Task 1: Contrastive Learning and Autoencoders for Semantic Textual Relatedness in Multilingual Texts

Udvas Basak* **Rajarshi Dutta*** **Shivam Pandey*** **Ashutosh Modi**
Indian Institute of Technology Kanpur (IIT Kanpur)
{udvasb20, rajarshi20, shivamp20}@iitk.ac.in
ashutoshm@cse.iitk.ac.in

Abstract

This paper describes our system developed for the SemEval-2024 Task 1: Semantic Textual Relatedness. The challenge is focused on automatically detecting the degree of relatedness between pairs of sentences for 14 languages including both high and low-resource Asian and African languages. Our team participated in two subtasks consisting of Track A: supervised and Track B: unsupervised. This paper focuses on a BERT-based contrastive learning and similarity metric based approach primarily for the supervised track while exploring autoencoders for the unsupervised track. It also aims on the creation of a bigram relatedness corpus using negative sampling strategy, thereby producing refined word embeddings.

1 Introduction

The semantic relatedness between texts in a language is fundamental to understanding meaning (Halliday and Hasan, 2014). Automatically detecting relatedness plays an essential role in evaluating sentence representations, question answering, and summarization (Abdalla et al., 2023). The fundamental difference between semantic similarity and relatedness is that semantic similarity only considers paraphrase or entailment relationships. In contrast, relatedness accounts for all commonalities between two sentences, e.g., topical, temporal, thematic, contextual, syntactic, etc. (Abdalla et al., 2023). As highlighted in Table 1, Sentences 1 and 2 are semantically similar, but sentences 2 and 3 would have low semantic similarity but high semantic relatedness.

In Track A (Task 1) of the Semantic Textual Relatedness (STR) task (Ousidhoum et al., 2024b), we are expected to calculate the degree of semantic relatedness between pairs of sentences in 14 different languages covering both African and Asian languages. Each pair of sentences is assigned a relatedness score in the range of 0 and 1. The major challenge lies in the efficient development of a metric to facilitate the calculation of the relatedness score between the sentence pairs and harnessing the structure of multiple languages to create an efficient model (Ousidhoum et al., 2024b). Our system is based on a contrastive learning approach, utilizing a composite lexical similarity-based measure for relatedness score calculation in the supervised task. Additionally, it involved the use of transformer autoencoders for the unsupervised task. We employed Distill-RoBERTa (Sanh et al., 2020) as the model for this purpose. Several other strategies were also tested within this framework, such as employing a Siamese architecture and retraining BERT with vocabulary expansion to incorporate tokens from additional low-resource languages. For the unsupervised task, the base model used to construct the denoising autoencoder was BERT-uncased (Devlin et al., 2019). The major challenge in this task was the devising and implementing data pre-processing schemes for diverse languages and various training methodologies. A number of experiments were conducted to come up with a unified metric for semantic relatedness calculations, which resulted in relatively better performances in various low-resource languages.[1]

#	Sentence
1	The mouse was chased by the cat in the yard.
2	The cat chased the mouse around the garden.
3	The dog barked loudly as the mouse scurried away.

Table 1: Difference between Similarity and Relatedness

2 Background

There have been several attempts to define and distinguish semantic relatedness from semantic simi-

* Equal Contributions

[1]The code can be found at https://github.com/Exploration-Lab/IITK-SemEval-2024-Task-1-Semantic-Relatedness

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1443–1448

larity. The basic metric used in these experiments is Spearman Rank Correlation. The correlation coefficient is calculated between the correctly annotated scores for the set of pairs of sentences and the scores returned by the models. Essentially, this removes the absolute values of the scores and focuses on the relative values and, hence, the relative relatedness between the pairs of sentences.

Initial experiments explored frequency measures such as lexical overlap (Shirude et al., 2021), related words, and related subjects and objects, leading to high Spearman correlations of 0.82 and 0.83 for BERTbase (mean) and RoBERTa-base (mean) on the CompLex dataset (Shardlow et al., 2020). Despite marginal improvements over a lexical overlap baseline, unsupervised models offer limited enhancement.

Normalized Google Distance(NGD) has been used as a novel metric for measuring semantic relatedness between words or concepts (Lopes and Moura, 2019), utilizing Google search result counts to quantify relatedness. NGD (Cilibrasi and Vitanyi, 2007) normalized counts considering the overall corpus size and the co-occurrence of terms in web pages.

Various approaches have been proposed for the Arabic language (Al Sulaiman et al., 2022) like automatic machine translation to translate English Semantic Textual Similarity (STS) data into Arabic, interleaving English STS data with Arabic BERT models, and employing knowledge distillation-based models to fine-tune them using a translated dataset. Multilingual knowledge distillation (Reimers and Gurevych, 2019) techniques have been proposed where a student model, $\hat{M}$, learns from a teacher model, M, on source language sentences and their translations by minimizing the mean-squared loss function. Focusing on low-resource Indian languages, a range of SBERT models has been introduced for ten popular Indian languages. IndicSBERT (Deode et al., 2023) utilized a two-step training method, fine-tuning models using the NLI dataset followed by Semantic Textual Similarity benchmarking (STSb), resulting in substantial improvements in embedding similarity scores and cross-lingual performance.

3 Dataset Description

The dataset (Ousidhoum et al., 2024a) consists of a total of 14 languages, namely Afrikaans, Algerian Arabic, Amharic, English, Hausa, Indonesian, Hindi, Kinyarwanda, Marathi, Modern Standard Arabic, Moroccan Arabic, Punjabi, Spanish, and Telugu. Every language consists of pairs of sentences with scores representing the degree of semantic textual relatedness between 0 and 1. The scores have been assigned to sentence pairs through a comparative annotation process (Ousidhoum et al., 2024a).

At the preliminary level, dataset length is the only non-semantic variable in these datasets. To assess semantic relatedness, it is crucial to mitigate these biases. Correlation coefficients between sentence lengths and scores fall in the range $-0.13 < \rho < 0.15$; hence, there is no discernible correlation between sentence lengths and scores, suggesting a well-distributed dataset suitable for training.

4 System Overview

Our baseline system of scoring a pair of sentences uses Jaccard Similarity, a lexical metric that calculates the number of token intersections over total tokens in both sentences. Our approach involves using Contrastive learning for the supervised part and auto-encoders for the unsupervised part. All these methods are discussed below, and our model supervised architecture is shown in Figure 1.

4.1 SimCSE

SimCSE (Gao et al., 2022), or Simple Contrastive Learning, is helpful in supervised and unsupervised learning, particularly in information retrieval, text clustering, and semantic tasks. This approach primarily uses Natural Language Inference (NLI) to create positive and negative sentence samples. It works by inducing slight variation in its representations through dropouts. The following step lies in aligning related sentences close in the embedding space and maximizing distances to unrelated sentences in each batch of data. For a supervised setting, it classifies positive samples as entailment pairs, while negative samples are derived from contradiction pairs. The training proceeds via minimizing the loss function: $-\log\left(\frac{e^{\frac{sim(h,h^+)}{\tau}}}{e^{\frac{sim(h,h^+)}{\tau}}+e^{\frac{sim(h,h^-)}{\tau}}}\right)$, where, h represents the current sentence and h^+ and h^- denotes the positive and negative samples respectively with τ being the temperature hyperparameter which controls the sensitivity and learning dynamics. The τ is mainly adjusted based on validation set perfor-

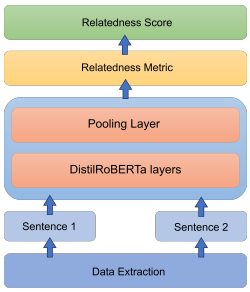


Figure 1: SIMCSE based approach for Track A

mance.

4.2 TSDAE

TSDAE (Wang et al., 2021) or Transformer Denoising through Auto Encoders is an elegant approach aimed at improving the quality of sentence embeddings through self-supervised learning. This method mainly uses sentence embeddings without labels, relying solely on the data structure to initiate learning. The first step lies in corrupting the existing tokens via methods like deleting random tokens, masking tokens, etc., and passing these modified representations through an encoder layer. The encoder layer outputs a dense latent representation, capturing the essence of the data in high-dimensional space. The encoded representations are then passed to a decoder, which attempts to reconstruct the original, uncorrupted sentences from the encoded representations. The decoder is typically another transformer model that has been trained to generate text based on the encoded embeddings. The main objective lies in minimizing the distance between the corrupted and reconstructed sentence representations through cross-entropy loss.

4.3 Training Scheme

For the supervised track, we used Distil-RoBERTa (Sanh et al., 2020) as the base model, which produced a vector representation for every word in the input sentence, resulting in a matrix of token embeddings. The embeddings were then fed to a pooling layer for the production of sentence-level embeddings, which finally proved to be useful for semantics-relatedness tasks. We used mean pooling because there were no dedicated [CLS] token representations for sequence classification tasks. As for the metrics, our approach was involved in designing a custom relatedness metric by combining standard distance-based metrics like cosine similarity, Mahalanobis distance, Euclidean and Manhattan distances, and lexical overlap-based metrics like Jaccard and Dice coefficients. For each pair of sentence embeddings in the dataset, we calculated these metrics. Not only did we calculate these metrics using the original embeddings, but we also calculated them after transforming the embeddings by raising them to higher powers (e.g., squaring them). These calculated metrics were then collected into a dataset, with each column named according to the metric and the power applied to the embeddings. For example, the column "Cosine Distance: 2" depicted the cosine distances between pairs of sentence embeddings after both embeddings in each pair have been squared. The dataset, therefore, finally had rows where each row was a 42-element vector. This vector encompassed the calculated metrics across different powers for the sentence embeddings. These enhanced sentence embeddings, with metrics covering higher orders, were then used to train the RoBERTa model. The goal was to produce scores that indicate how related different sentences are across various languages. The libraries used are in Table 5.

5 Experiments

5.1 Supervised Task

Static Approaches: The baseline models, Jaccard Coefficient, Dice Coefficient, and similar coefficients after removing stopwords were calculated to arrive at reliable baseline metrics to build upon.
Multilingual BERT: Since the best-performing model for English involved BERT, an attempt was made to train multilingual BERT: mBERT (Pires et al., 2019), by extending the vocabulary to allocate the tokens of various low-resource languages like Amharic, Hausa, Algerian Arabic, Afrikaans, Indonesian etc. The approach included generation of the vocabulary of each of the languages from the training data and then calling the pre-trained mBERT model and tokenizer. The trained tokenizer was extended to include the new tokens generated from the vocab of the corresponding low resource languages. A trainable feed-forward network was added with the corresponding dropout. The loss metric used was mean squared error loss on both the training validation data and the Spearman rank correlation were calculated at the end of each validation epoch. Finally, finetuning multilingual BERT yielded considerably good results and this avenue was found suitable for exploration, especially for the cross-lingual task.

The relatedness metric was approximated using a

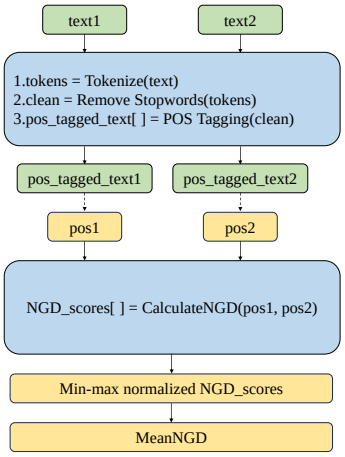


Figure 2: NGD Calculation flowchart

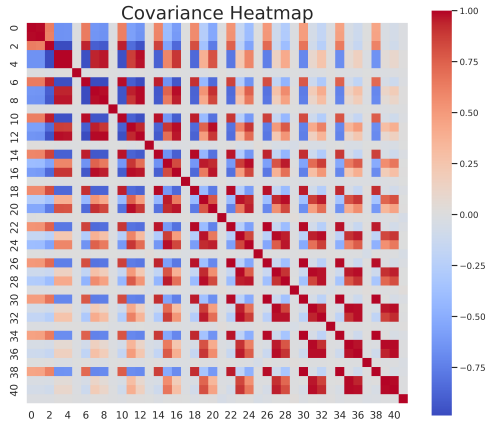


Figure 3: Covariance Matrix between all 42 metrics

trainable feed forward layer by experimenting with the number of hidden layers and activations. It was observed that having 3 hidden layers resulted in fairly good relatedness scores between pairs of sentences in almost all the languages. It was observed that GeLU performed better than ReLU activation primarily due to its steep curve around 0 which helps to model complex functions better. The combination of learning rate and weight decay also resulted in a stable training curve, avoiding sub-optimal loss convergence. Thus these specific hyperparameters were optimal for mBERT retraining in terms of resource constraints and performance. The corresponding hyperparameters of the best performing model is presented in App. Table 4.

Contrastive Learning: The details of the system are described in §4.1. Experiments were run on the number of epochs during training.

Combined Similarity Metric: Normalized Google Distance (Cilibrasi and Vitanyi, 2007) calculates a relatedness metric for two input sentences. It was proposed as a strong metric for relatedness by Lopes and Moura (2019). It starts by tokenizing and removing stop words from both sentences, followed by part-of-speech tagging. Then, it calculates NGD values for pairs of words with the same part of speech in both sentences. The NGD scores are normalized and averaged to compute the overall NGD score, representing the degree of relatedness between the two sentences. The flowchart for the process is shown in Figure 2. Cosine similarity is the standard metric used to find similarity between two sentence embeddings, which gives a 0.81-0.82 baseline score for this problem.However, similar or better results can be seen when other distance metrics like Mahalanobis Distance(0.82) and Euclidean Distance(0.83) are observed between the embeddings. Further, augmenting this with more direct relatedness metrics like NGD is promising for better results. A simple supervised deterministic regression model can be implemented to combine these metrics. Furthermore, to explore the importance of each of these metrics, a simple covariance matrix (Figure 3) can show how the vector metrics on higher element-wise-powered vectors hold information not caught directly at the lower powers of the vectors.

To implement this supervised regression model, a simple 3-layered feed-forward neural network (with neurons **[25]+[50]+[25]**) is trained with. The layers are chosen to construct a lightweight network. Each data feature $\mathbf{x}$ was composed as:

$$\mathbf{x_i} = \{S(v_{i,1}, v_{i,2}), S(v_{i,1}^2, v_{i,2}^2), \ldots, S(v_{i,1}^{10}, v_{i,2}^{10}), J(v_{i,1}, v_{i,2}), D(v_{i,1}, v_{i,2})\}$$

where $v^i = (v_1^i, v_2^i, \ldots v_n^i)$

$S(a,b) = \{C(a,b), E(a,b), M_1(a,b), M_2(a,b)\}$

$C(a,b) =$ Cosine Similarity between a and b

$E(a,b) =$ Euclidean Distance between a and b

$M_1(a,b) =$ Manhattan Distance between a and b

$M_2(a,b) =$ Mahalanobis Distance between a and b

$J(a,b) =$ Jaccard similarity between a and b

$D(a,b) =$ Dice similarity between a and b

Even though this metric did not show promise in English, this was helpful in some low-resource languages, and hence was part of our system design for some languages.

5.2 Unsupervised Task

Bigram Corpus Creation and Training Process: We developed a pipeline to generate a bigram dataset from any language corpus. A three-part tuple was created for every bigram found to note how

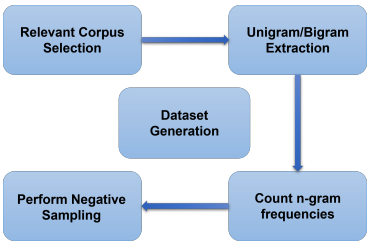


Figure 4: Bigram Corpus Creation Flowchart

often it appeared within the same sentence, paragraph, and entire document. This process aimed to quantify the connections between words by tracking their repeated sentence occurrences. Our main objective was to use these co-occurrence frequencies to produce word embeddings. We planned to enhance our method by applying hierarchical clustering, which helps identify word similarities and relationships. Moreover, we decided to use a 1:1 **negative sampling strategy** to refine the embeddings further. These embeddings were intended for computing relatedness scores, leveraging bigrams derived from sentence pairs and their lexical overlaps. Figure 4 presents a diagram illustrating this process.

TSDAE: A pipeline was developed to implement TSDAE (refer §4.2) on the languages. The number of epochs was changed and experimented on for various languages. Overall, around 20-25 epochs resulted in good results. No weight decay was implemented, and a learning rate of 3e-5 was used.

6 Results

The results for the supervised and unsupervised are mentioned in Table 2 and Table 3. The leaderboard highlights that the chosen contrastive learning approach did not perform well, especially for some languages where it fell significantly below the baseline scores provided by their system, such as Hausa, Moroccan Arabic, Telugu, etc. Some possible shortcomings of this approach might be that the negative samples were not distinguishable enough from the positive samples, which might also be attributed to the poor performance of the transformer models on languages, especially with complex lexical structures. The other issue might be the traditional loss function, which might not be good enough to capture the degree of semantic relationships between sentences. For the unsupervised track, our approach is performing reasonably well for most languages, which is indicated by the correlation score being more significant than the baseline provided.

Language	Rank	Score	Baseline Score
Amharic	17	0.55	0.85
Hausa	21	0.22	0.69
Kinyarwanda	21	0.14	0.72
Moroccan Arabic	22	0.36	0.77
Spanish	23	0.59	0.7
Algerian Arabic	23	0.34	0.6
Marathi	24	0.67	0.88
Telugu	25	0.28	0.82
English	31	0.81	0.83

Table 2: Evaluation Phase Results in Codalab Leaderboard for Track A

Language	Rank	Score	Baseline Score
Algerian Arabic	2	0.49	0.43
English	4	0.81	0.68
Amharic	6	0.07	0.72
Hausa	6	0.38	0.16
Moroccan Arabic	6	0.36	0.27
Spanish	9	0.59	0.69

Table 3: Evaluation Phase Results in Codalab Leaderboard for Track B

7 Error Analysis

After the evaluation phase, we were provided with the labels for the evaluation data. On experimenting with the semantic relatedness scores for some languages, mainly **Hausa** and **Kinyarwanda**, we found out that our system was not performing well enough on these languages even after subsequent training and hyperparameter optimizations. The issue would be primarily attributed to the complex lexical variations and grammar rules of these languages. As for the unsupervised track, generating a bigram corpus for the case of **Amharic** seemed difficult due to its language structure.

8 Conclusion

By utilizing various approaches like contrastive learning, autoencoders, a custom relatedness metric incorporating all of the available lexical similarity metrics, we have developed a system capable of evaluating the degree of semantic relatedness between pairs of sentences in diverse high and low resource languages. In future, we will study the properties of each low resource language to find out where the models are performing poorly than relying too much on pre-trained models. This would give much clearer insights into semantics of each language thus improving the overall efficiency and performance of our system.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study.

Mansour Al Sulaiman, Abdullah M Moussa, Sherif Abdou, Hebah Elgibreen, Mohammed Faisal, and Mohsen Rashwan. 2022. Semantic textual similarity for modern standard and dialectal arabic using transfer learning. *Plos one*, 17(8):e0272991.

Rudi L. Cilibrasi and Paul M.B. Vitanyi. 2007. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023. L3cube-indicsbert: A simple approach for learning cross-lingual sentence representations using multilingual bert. *arXiv preprint arXiv:2304.11434*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. 2014. *Cohesion in english*. 9. Routledge.

Carla Teixeira Lopes and Diogo Moura. 2019. Normalized google distance in the identification and characterization of health queries.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert?

Nils Reimers and Iryna Gurevych. 2019. Making monolingual sentence embeddings multilingual using knowledge distillation. *CoRR*, abs/1908.10084.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — a new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

Neil Shirude, Sagnik Mukherjee, Tushar Shandhilya, Ananta Mukherjee, and Ashutosh Modi. 2021. IITK@LCP at SemEval-2021 task 1: Classification for lexical complexity regression task. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 541–547, Online. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning.

A Appendix

Hyperparameters	Values
Learning Rate	2e-5
Dropout	0.1
Weight Decay	0.01
Number of Linear Layers	3
Activation	GELU
Max Length	512

Table 4: Hyperparameters for mBERT retraining

Libraries	Version
numpy	1.25.2
PyTorch	2.0.1+cu117
transformers	4.36.2
sentence_transformers	2.2.2
scikit-learn	1.3.2
pandas	2.1.4

Table 5: Libraries used in our system

Compos Mentis at SemEval2024 Task6: A Multi-Faceted Role-based Large Language Model Ensemble to Detect Hallucination

Souvik Das , Rohini K. Srihari
{souvikda, rohini}@buffalo.edu
Department of Computer Science and Engineering, University at Buffalo, NY.

Abstract

Hallucinations in large language models (LLMs), where they generate fluent but factually incorrect outputs, pose challenges for applications requiring strict truthfulness. This work proposes a multi-faceted approach to detect such hallucinations across various language tasks. We leverage automatic data annotation using a proprietary LLM, fine-tuning of the Mistral-7B-instruct-v0.2 model on annotated and benchmark data, role-based and rationale-based prompting strategies, and an ensemble method combining different model outputs through majority voting. This comprehensive framework aims to improve the robustness and reliability of hallucination detection for LLM generations. Code and data[1]

1 Introduction

The modern natural language generation (NLG) (OpenAI et al., 2023; Touvron et al., 2023) landscape faces two interconnected challenges: firstly, current neural models have a tendency to produce fluent yet inaccurate outputs, and secondly, our evaluation metrics are better suited for assessing fluency rather than correctness(Bang et al., 2023; Guerreiro et al., 2023). This phenomenon, known as "hallucination," (Ji et al., 2023) where neural networks generate plausible-sounding but factually incorrect outputs, is a significant hurdle, especially for NLG applications that require strict adherence to correctness. For instance, in machine translation(Lee et al., 2019), producing a fluent translation that deviates from the source text's meaning renders the entire translation pipeline unreliable. This issue may arise as LLMs are trained on vast amounts of data from the internet, which can contain inaccuracies, biases, and false information. Also, it may arise due improper representations learned during training even if good quality data is used. As a result, LLMs can sometimes hallucinate or fabricate details, especially when prompted to discuss topics outside their training data or make inferences beyond their capabilities.

Hallucination detection (Liu et al., 2022), also known as factual verification or truthfulness evaluation, identifies and mitigates these hallucinations in the outputs of LLMs. This is an active area of research and development, as it is crucial for ensuring the reliability and trustworthiness of LLM-generated content, particularly in high-stakes domains such as healthcare, finance, and legal applications. In this task, the primary focus will be to classify whether a generation is hallucinated.

This work proposes a multi-faceted approach to detecting hallucinations in large language models' outputs. We employ automatic data annotation using a proprietary LLM (Claude 2.1[2]) to label examples from the provided training set as hallucinated or not. Then we fine-tune the Mistral-7B-instruct-v0.2[3] model on this annotated data as well as the HaluEval benchmark (Li et al., 2023) to create two fine-tuned models. To improve performance, we use role-based prompting that casts the task in specific contexts like fact-checking. We also leverage rationale-based prompting, asking the LLM to justify its hallucination label. Finally, an ensemble method combines outputs from the fine-tuned Mistral models, Claude 2.1, and different prompting strategies via majority voting. This comprehensive approach aims to enhance the robustness and reliability of hallucination detection across various language tasks.

2 Task Details

This shared task (Mickus et al., 2024) aims to foster the growing interest within the community in ad-

[1] https://github.com/souvikdgp16/shroom_compos_mentis

[2] https://www.anthropic.com/news/claude-2

[3] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1449–1454
June 20-21, 2024

dressing this issue. Participants are tasked with performing binary classification to identify instances of fluent overgeneration hallucinations in two different setups: a model-aware track and a model-agnostic track. Essentially, participants must detect grammatically sound outputs that contain incorrect or unsupported semantic information, inconsistent with the source input, with or without having access to the model that produced the output.

To facilitate this task, participants are provided with a collection of checkpoints, inputs, references, and outputs from systems covering three different NLG tasks: definition modeling (DM), machine translation (MT), and paraphrase generation (PG). These systems will be trained with varying degrees of accuracy. The validation and test sets will include binary annotations from at least five annotators, with a majority vote determining the gold label.

2.1 Data

The data split is shown in Table 1.

Task	Validation	Test
model agnostic	500	1500
model aware	500	1500

Table 1: Data-split statistics.

Each data split file is formatted as a JSON list. Each element in this list corresponds to a data point as shown:


```
{
        "hyp": "(uncountable) The study of trees.",
        "ref": "tgt",
        "src": "It is now generally supposed that the
forbidden fruit was a kind of citrus , but certain facts
connected with  arborolatry  seem to me to disprove this
opinion .",
        "tgt": "The worship of trees.",
        "model": "",
        "task": "DM",
        "labels": [
            "Hallucination",
            "Hallucination",
            "Hallucination"
        ],
        "label": "Hallucination",
        "p(Hallucination)": 1.0
}
```


Each data instance contains the following key elements: a task (task) indicating the language model's objective; a source (src) input; a target reference (tgt); a hypothesis (hyp) which is the model's actual output; a set of per annotator hallucination labels (labels); a majority-based gold hallucination label (label); and a probability score (p(Hallucination)) representing the proportion of annotators who labeled the instance as hallucinated.

2.2 Evaluation Protocol

Submissions are evaluated using two criteria:

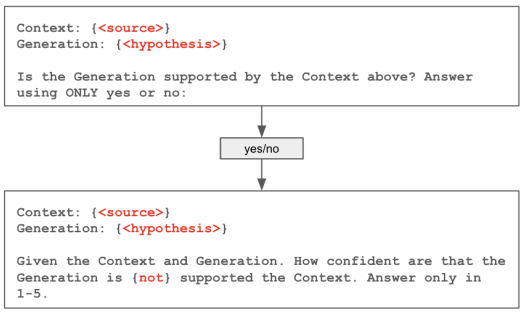
1. **Accuracy**: the system accuracy reached on the binary classification.

2. ρ: the Spearman correlation of the systems' output probabilities with the proportion of the annotators marking the item as overgenerating.

3 System Description

3.1 Automatic Data Annotation

We automatically annotate the unlabeled training data provided by the organizers. We use a strong proprietary Large Language Model(LLM) Claude 2.1 to annotate the data automatically. Since, annotations from Claude 2.1 might not be fully reliable we use a confidence-based measure to select only those training examples where the LLM is confident enough. We use the following prompts:



First, we prompt the LLM to get the hallucination label. Then we again prompt the LLM to do a retrospect on the decision it has made by asking it how confident it is with the decision. We filter out all the examples with a score less than 5.

3.2 Fine-tuning Mistral-7B-instruct-v0.2

We train two fine-tuned versions of Mistral-7B-instruct-v0.2 for this task:
Fine-tuned on our data: We split our automatically annotated dataset in 8:1:1 split for training, validation and testing. We adopt a generative approach for classification where the instruction was fed in this fashion: `[INST]`*prompt*`[/INST]`, where the *prompt* is the same as it is used during annotation phase. The goal is to generate the hallucination label. The test F1-score was 82.03%. We name this model as Mistral-7B-instruct-v0.2-halu-internal.
Fine-tuned on HaluEval dataset: Hallucination Evaluation benchmark for Large Language Models (HaluEval), a large collection of generated and

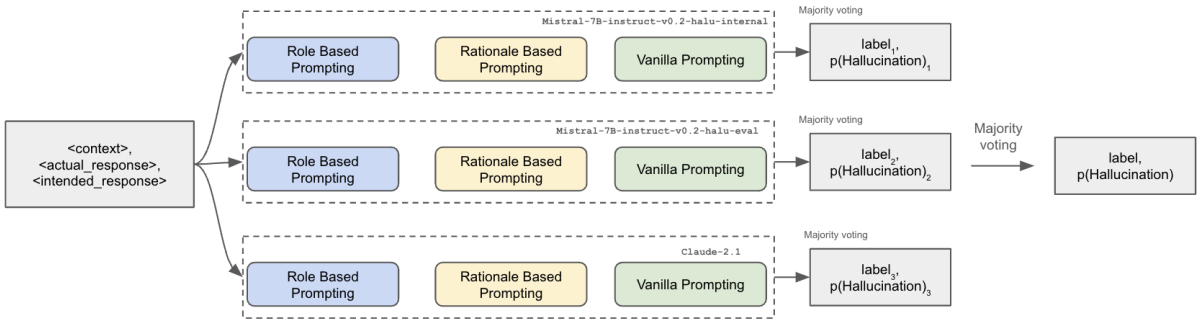


Figure 1: Our overall ensemble-based inference pipeline. We use majority voting at the model level and overall pipeline level to determine the final hallucination label.

human-annotated hallucinated samples for evaluating the performance of LLMs in recognizing hallucination HaluEval dataset contains $30,000$ hallucinated samples with $10,000$ examples for each task of QA, dialogue, and summarization. Here also, we adopt a generative approach for classification with the same instruction sequence as used during fine-tuning using our data. The test F1-score was 77.95%. We name this model as Mistral-7B-instruct-v0.2-halu-eval.

Hyperparameters: We use the original weights of Mistral-7B-instruct-v0.2 released by Mistral AI. We use QLoRA(Dettmers et al., 2023) for parameter-efficient fine-tuning. We set the maximum length of the input sequence to 512 and the rank k and α in QLoRA to 16 and 8, respectively. We use the `bitsandbytes` library to initialize the QLoRA parameters. We use an 8-bit Paged Adam optimizer to update QLoRA parameters with a batch size of 64 and learning rates of 1e-7. The trainable QLoRA parameters ($\sim$ 19.5M) are fine-tuned on 2 NVIDIA A5000-24GB GPUs. All the hyperparameter are tuned using the provided trial data, k and α were varied in the range of [4,16] with a step of 4, batch size was varied in the range of [32,72] with a step of 16, and the learning rate was varied from 1e-8 to 1e-7, the best performing hyperparameters are reported.

3.3 Role Based Prompting

Since we are dealing with multiple tasks, the same prompt might not be suitable for all the tasks during inference. We create task-specific role-based prompt for each task using the following prompt template:

```
Given the <intended_response>, <context> and
<actual_response> :

<intended_response>: {tgt}.
<context>: {src}.
<actual_response>: {hyp}.

{ROLE}

State whether the <actual_response> supports <context> and
<intended_response>. Answer using ONLY yes or no:
```

`<intended_response>` is the golden response, `<actual_response>` is the actually generated response. Here the inference-time roles will be based on the following Table:

	Role
Definition Modelling	Imagine yourself as a fact-checker; your job is to check whether <actual_response>is the definition of <context>.
Paraphrase Generation	Imagine yourself as a paraphrase-checker; your job is to check whether <actual_response> is an actual paraphrase of <context>. That means the meaning of <actual_response> should be the same as <context>, however <actual_response> will contain lesser words than <context>.
Machine Translation	Imagine yourself as a translation-checker; your job is to check whether <actual_response> is an actual translation of <context>.

Table 2: Role Definitions.

3.4 Rationale Based Prompting

We notice that when LLMs are prompted to produce rationale for its decision it often elicits more truthful response. Due to this observation we prompt the LLM to generate the explanation classifying the generation is hallucinated or not. The prompt is as follows:

```
Given the <intended_response>, <context> and
<actual_response> :

<intended_response>: {tgt}.
<context>: {src}.
<actual_response>: {hyp}.

{ROLE}

Come up with an explanation of whether the <actual_response>
supports <context> and <intended_response>.
Using the explanation, choose a final answer between yes or
no.
The final answer should be in this format:
{"explanation":<explanation>, "final_answer":<yes or no>}
```

3.5 Inference Ensemble

We combine all our prompting strategies to simulate an annotator for each sample. Also, we create an ensemble of three models: (1) Mistral-7B-instruct-v0.2-halu-internal (2) Mistral-7B-instruct-v0.2-halu-eval (3) Claude 2.1. Along with the role-based and rationale-based prompting we also incorporate a vanilla prompting where we just ask the LLM to come up with the hallucination label without assuming any role or generating a rationale, like this:

```
Given the <intended_response>, <context> and
<actual_response> :

<intended_response>: {tgt}.
<context>: {src}.
<actual_response>: {hyp}.

State whether the <actual_response> supports <context> and
<intended_response>. Answer using ONLY yes or no:
```

For each model pipeline, we get 3 hallucination labels; the pipeline label is the most common label out of 3. The hallucination probability score is determined by this equation: $p(\text{Hallucination}) = \frac{\#\text{halluciation\_labels}}{3}$. We get the hallucination label and $p(\text{Hallucination})$ for the three pipelines, and again we do a majority voting to get the final hallucination label. The final $p(\text{Hallucination})$ is set to the maximum probability of the selected hallucination label across the pipeline. We use greedy decoding for the Mistral-based models with a temperature of 0.8. Average cost of running Claude APIs for each is about 7$ for validation set and 16$ for test set. For Claude inference we use a temperature of 0.9.

4 Results

Table 3 and 4 show the results for model-aware and model-agnostic hallucination detection tasks for validation split. For both cases, we notice increased performance with rationale-based prompts for all the models. Subsequently, our ensemble-based pipeline boosts the performance even more. On the other hand, the performance of the HaluEval fine-tuned dataset is superior to our annotated dataset because there is a large possibility of noise getting introduced during our annotation process. Our annotation process uses verbalized model confidence as a proxy for data filtration; if the model is not calibrated correctly, this might lead to a faulty filtration process.

Configuration	Prompting Technique	Accuracy	Rho
Mistral-7B-instruct-v0.2-halu-internal	role-based	0.711	0.562
Mistral-7B-instruct-v0.2-halu-eval	role-based	0.724	0.588
Claude2.1	role-based	0.723	0.563
Mistral-7B-instruct-v0.2-halu-internal	rationale-based	0.724	0.566
Mistral-7B-instruct-v0.2-halu-eval	rationale-based	0.73	0.564
Claude2.1	rationale-based	0.728	0.566
Mistral-7B-instruct-v0.2-halu-internal	vanilla	0.712	0.562
Mistral-7B-instruct-v0.2-halu-eval	vanilla	0.72	0.553
Claude2.1	vanilla	0.712	0.565
3-model-ensemble	all	0.738	0.568

Table 3: Validation results for model-agnostic task.

Configuration	Prompting Technique	Accuracy	Rho
Mistral-7B-instruct-v0.2-halu-internal	role-based	0.713	0.568
Mistral-7B-instruct-v0.2-halu-eval	role-based	0.733	0.576
Claude2.1	role-based	0.723	0.556
Mistral-7B-instruct-v0.2-halu-internal	rationale-based	0.726	0.567
Mistral-7B-instruct-v0.2-halu-eval	rationale-based	0.723	0.569
Claude2.1	rationale-based	0.731	0.572
Mistral-7B-instruct-v0.2-halu-internal	vanilla	0.708	0.562
Mistral-7B-instruct-v0.2-halu-eval	vanilla	0.723	0.533
Claude2.1	vanilla	0.726	0.566
3-model-ensemble	all	0.736	0.579

Table 4: Validation results for model-aware task.

Task	Configuration	Accuracy	Rho
model agnostic	3-model-ensemble	0.738	0.595
model aware	3-model-ensemble	0.756	0.566

Table 5: Evaluation results.

During evaluation, we ran our best-performing pipeline i.e., the ensemble of 3 models. A performance similar to the validation set is observed here. Our team ranked 33 out of 48 for model agnostic sub-task and 29 out of 45 for model aware sub-task.

5 Conclusion

This work proposes a multi-faceted approach for detecting hallucinations in large language model outputs across various natural language tasks. It employs automatic data annotation, fine-tuning state-of-the-art models on annotated data and benchmarks, role-based and rationale-based prompting strategies, and an ensemble method combining multiple model outputs. The ensemble pipeline achieves promising results on model-agnostic and model-aware evaluation settings for hallucination detection. While challenges remain, this comprehensive framework highlights the potential of carefully designed prompting, model fine-tuning, and ensembling techniques to enhance the robustness and reliability of factual verification in language model generations, paving the way for developing more trustworthy natural language generation systems.

Acknowledgements

We thank the anonymous reviewers for providing valuable feedback on our manuscript. This work is partly supported by NSF grant number IIS-2214070. The content in this paper is solely the responsibility of the authors and does not necessarily represent the official views of the funding entity.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. Hallucinations in large multilingual translation models.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2019. Hallucinations in neural machine translation.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan

Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

NYCU-NLP at SemEval-2024 Task 2: Aggregating Large Language Models in Biomedical Natural Language Inference for Clinical Trials

Lung-Hao Lee[1,2], **Chen-Ya Chiou**[2] **and Tzu-Mi Lin**[1]
[1]Institute of Artificial Intelligence Innovation, National Yang Ming Chiao Tung University
[2]Department of Electrical Engineering, National Central University
{lhlee, ltmdegf4.ii12}@nycu.edu.tw, 109501528@cc.ncu.edu.tw

Abstract

This study describes the model design of the NYCU-NLP system for the SemEval-2024 Task 2 that focuses on natural language inference for clinical trials. We aggregate several large language models to determine the inference relation (i.e., entailment or contradiction) between clinical trial reports and statements that may be manipulated with designed interventions to investigate the faithfulness and consistency of the developed models. First, we use ChatGPT v3.5 to augment original statements in training data and then fine-tune the SOLAR model with all augmented data. During the testing inference phase, we fine-tune the OpenChat model to reduce the influence of interventions and fed a cleaned statement into the fine-tuned SOLAR model for label prediction. Our submission produced a faithfulness score of 0.9236, ranking second of 32 participating teams, and ranked first for consistency with a score of 0.8092.

1 Introduction

Biomedical Natural Language Inference (NLI) seeks to determine whether a proposed statement is entailment, contradiction, or neutral according to a given clinical trial. The MEDIQA-2019 shared task (Ben Abacha et al., 2019) covered an NLI subtask in the medical domain, including clinical sentences from the MIMIC-III database (Romanov and Shivade, 2018). In this shared task, most systems were built on the BERT model (Devlin et al., 2019) and MT-DNN (Liu et al., 2019). The BERT-BiLSTM-Attention model (Lee et al., 2019) was proposed for medical text inference. The DoubleTransfer model (Xu et al., 2019) was presented to use a multi-source transfer learning approach to acquire knowledge from MT-DNN and Sci-BERT (Beltagy et al., 2019). In addition, since the evaluation data is sourced from the clinical domain, variations of BERT such as BioBERT (Lee et al., 2020) and ClinicalBERT (Huang et al., 2020) were used frequently.

SemEval-2023 Task 7 (Jullien et al., 2023b) (called NLI4CT) focused on multi-evidence natural language inference for Clinical Trial Reports (CTR) (Jullien et al., 2023a). Participants should determine the inference relation (i.e., entailment or contradiction) between CTR-statements in the NLI subtask. The sentence-level and token-level encodings were exploited in a multi-granularity inference network (MGNet) (Zhou et al., 2023). The DeBERTa-v3 model (He et al., 2023) was fine-tuned on the prompted input sentences to discriminate the inference relation between the statement and clinical trials (Wang et al., 2023b). The BioLinkBERT transformer (Yasunaga et al., 2022) was used with a soft voting ensemble mechanism to enhance the NLI performance (Chen et al., 2023). The Flan-T5 model (Chung et al., 2022) was fine-tuned with instructions to explore its capabilities for multi-evidence NLI (Kanakarajan and Sankarasubbu, 2023).

Following the success of the NLI4CT-2023 task, SemEval-2024 Task 2 (Jullien et al., 2024) re-grounds this task in interventional and causal analyses of NLI models (Yu et al., 2022), with a contrast set containing the designed interventions and expected labels to investigate the faithfulness and consistency of the developed models. This task is based on the same collection of breast cancer CTRs (Jullien et al., 2023a). The statements in the training set are identical to those in the previous task, but perform a variety of interventions to statements on the development and test sets, making claims about a single CTR or comparing

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1455–1462

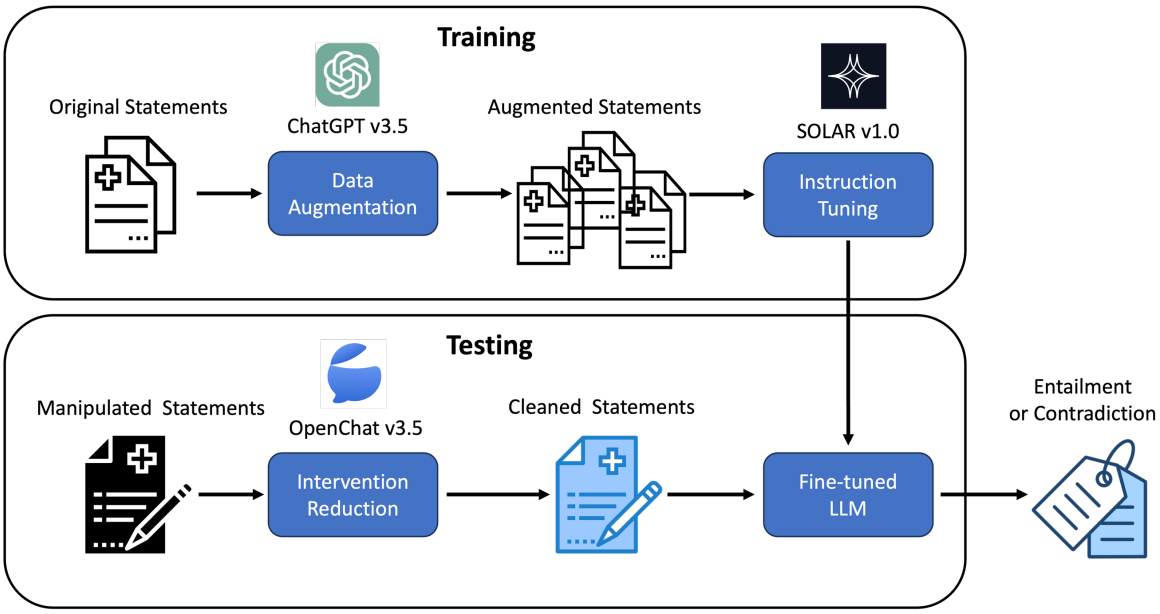


Figure 1: Our NYCU-NLP system architecture for the NLI4CT-2024 task.

two CTRs while either preserving or inversing the entailment relations. For the NLI4CT-2024 task, given a statement with/without interventions, the participating system should determine the inference relation as either entailment or contradiction.

This paper describes the **NYCU-NLP** (**N**ational **Y**ang Ming **C**hiao Tung **U**niversity, **N**atural **L**anguage **P**rocessing Lab) system for the NLI4CT-2024 task. Given the promising results obtained by Large Language Models (LLM) for various NLP tasks, we aggregate several LLMs in biomedical NLI for clinical trials. We use ChatGPT (OpenAI, 2023) to augment original statements and then fine-tune the SOLAR model (Kim et al., 2023) with instructions designed for the NLI task. Since a statement may be manipulated during testing inference phase, we first fine-tune the OpenChat model (Wang et al., 2023a) to reduce the influence of interventions. Finally, a cleaned statement along with CTRs is fed into the fine-tuned SOLAR model for label prediction (i.e., entailment or contradiction). Evaluation results show that our proposed NYCU-NLP system had a faithfulness score of 0.9236, ranking second among 32 participating teams, and ranked first for consistency with a score of 0.8092.

The rest of this paper is organized as follows. Section 2 describes the NYCU-NLP system for the NLI4CT-2024 task. Section 3 presents the results and performance comparisons. Conclusions are finally drawn in Section 4.

2 The NYCU-NLP System

Fig. 1 shows our NYCU-NLP system architecture for the NLI4CT-2024 task. Our system is composed of four main parts: 1) ChatGPT (OpenAI, 2023) for data augmentation; 2) Instruction tuning on SOLAR (Kim et al., 2023); 3) OpenChat (Wang et al., 2023a) for intervention reduction; and 4) Fine-tuned LLM for label prediction.

2.1 Data Augmentation

We use ChatGPT (OpenAI, 2023) to augment the training data for intervention adaptation. Fig. 2 shows the prompts inputted to the ChatGPT API (gpt-3.5-turbo-1106) and example outputs. We provide a system prompt to set up ChatGPT as a writer, skilled in rewriting sentences. For the first prompt, we obtain three rewritten statements without any restrictions. For the second prompt, we ask the LLM to rephrase four statements, each independently fulfilling the following requirements: 1) change as many words as possible except the existing words in primary and secondary trials; 2) change the order of existing words in the statement; 3) change percentage numbers to decimals and vice versa; and 4) change the numbers in terms of percentages and decimals to a

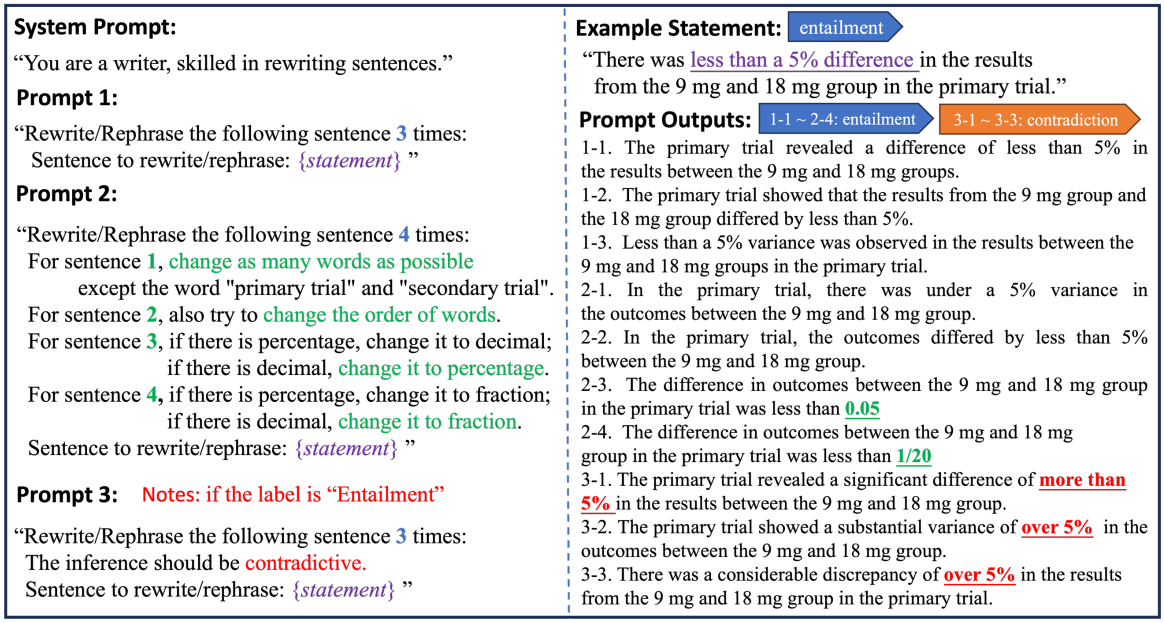


Figure 2: Prompts designed for data augmentation in ChatGPT v3.5.

fraction form. The above prompts are used for both entailment and contradiction labels. However, the third prompt is designed for the entailment label only, rewriting the original statements with contrary meanings to obtain contradictive inferences.

We also clean augmented statements to remove potentially inappropriate statements. For prompts 2-3 and 2-4, if the original statements do not contain numbers, but augmented statements contain numbers in any forms, we remove those augmented statements because these numbers are mostly hallucinations.

2.2 Instruction Tuning

We use original and augmented statements with the corresponding labels to fine-tune the SOLAR model (Kim et al., 2023). SOLAR-10.7B presents a depth up-scaling (DUS) technique to integrate Mistral 7B (Jiang et al., 2023) weights into the upscaled layers, and performs continued pre-training for the entire model. Supervised fine-tuning (SFT) and direct preference optimization (DPO) (Rafailov et al., 2023) were then used to fine-tune the model with designed instructions.

We continually fine-tune the SOLAR-10.7B-Instruct-v1.0 LLM. We use instruction tuning (Wei et al., 2022) and LoRA (Hu et al., 2021) techniques with prompts shown in Fig. 3 to optimize the SOLAR model for this NLI task. Flash attention

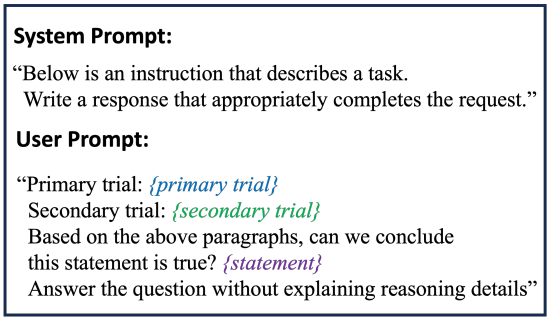


Figure 3: Prompts used for instruction tuning

(Dao et al., 2022) is also used to reduce the GPU requirements and accelerate the model fine-tuning process.

2.3 Intervention Reduction

A testing statement may be manipulated with some interventions, including numerical reasoning, vocabulary and syntax, and semantics, to investigate the consistency and faithfulness of the developed models. Technical details used to perform the interventions were not disclosed during the evaluation phase.

Therefore, we fine-tuned OpenChat v3.5 (Wang et al., 2023a) to reduce the influence of interventions. OpenChat is a framework used to advance open-source language models with mixed-quality data. As shown in Fig. 4, we used two exemplars for two-shot prompt learning. First, we

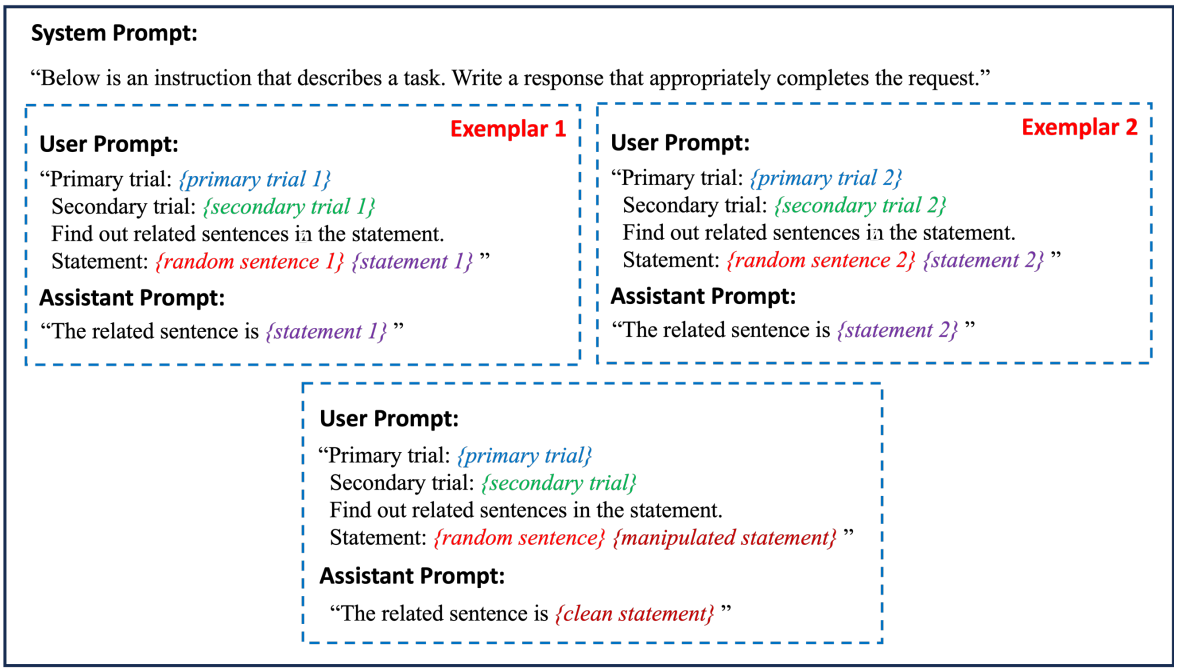


Figure 4: Prompts designed for intervention reduction in OpenChat v3.5.

randomly collected 10,000 abstracts published from Jan. 1st to Jan. 10th, 2024 from the arXiv preprint server. These were segmented into a total of 21,135 sentences. Finally, we randomly selected one sentence as an intervention sentence for both exemplars. The statements were selected from the training set, in which the first statement contains only one sentence and the second statement contains at least two sentences.

During the evaluation phase, an original statement is regarded as a manipulated statement and the fine-tuned LLM is expected to identify a cleaned statement. In most cases, a cleaned statement is a part of an original statement for intervention reduction. If an output statement contains sentences that don't belong to the original statement, the cleaned statement will be discarded and the original statement is used as the input for inference testing.

2.4 Fine-tuned LLMs for Label Prediction

Following the instruction shown in Fig. 3, the fine-tuned LLM processes a given statement based on the CTRs and answers the question without explaining its reasoning in detail. In the LLM response, if the first token is Yes or True, the predicted label is entailment, and otherwise contradiction. If the first token belongs to neither of these characteristics, we will check the vocabulary table to determine the corresponding probabilities of Entailment and Contradiction tokens. If the former exceeds the latter, the predicted label is returned as entailment, and otherwise contradiction.

3 Experiments and Results

3.1 Data

The datasets were mainly provided by task organizers (Jullien et al., 2024). A total of 1000 collected breast cancer CTRs were used as known premises. The training set used 1,700 statements to make claims about a single CTR or to compare two CTRs labelled as either entailment or contradiction. We used these statements for data augmentation, producing a total of 13,484 generated statements for LLM fine-tuning.

During the system development and evaluation phases, task organizers performed a variety of interventions on the statements in the development and test sets, either preserving or inversing the entailment relations. A total of 2,142 statements were used to develop the system and obtain the optimized parameters. Finally, the test set containing 5,500 statements was used to evaluate the system performance.

3.2 Settings

In addition to our fine-tuned SOLAR model (Kim et al., 2023), we used Mistral (Jiang et al., 2023),

Model (#para)	Development			Test		
	F1	Faithfulness	Consistency	F1	Faithfulness	Consistency
Orca2 (13B)	0.8223	0.8899	0.7914	0.7747	0.8692	0.7643
Qwen (14B)	0.8367	0.8542	0.8076	0.7657	0.8681	0.7730
Mistral (7B)	0.8500	0.9196	0.8213	0.7623	0.8611	0.7805
SOLAR (10.7B)	**0.8842**	**0.9554**	**0.8506**	**0.7790**	**0.9236**	**0.8092**

Table 1: Fine-tuned LLM results for the development and test sets.

Orca2 (Mitra et al., 2023) and Qwen (Bai et al., 2023) LLMs for performance comparison. All models were downloaded from HuggingFace[1]. We continuously fine-tuned these models using the augmented training set. All models were configured to obtain the highest average faithfulness and consistency scores on the development set. The hyperparameter values of our used SOLAR LLM were finally optimized as follows: epochs 20; batch size 8; optimizer Adafactor; learning rate schedule used a cosine decay with optional warmup; warmup ratio 0.05; max learning rate 7.5e-5; LoRA r 16; LoRA alpha 16; LoRA drop 0.05; max token length 2048 and original statement sample ratio 0.3.

3.3 Metrics

The *control F1* measures fundamental model performance of those testing instances without interventions, identical to the previous NLI4CT-2023 task and thus facilitating a direct performance comparison.

Faithfulness is estimated to measure the model's ability to correctly change its predictions when exposed to a semantic-altering intervention. The better system is expected to make the correct prediction for the correct reason.

Consistency measures the model's ability to predict the same label for original statements and contrast statements for semantic-preserving interventions. The better system is expected to produce the same outputs for semantically equivalent problems.

3.4 Results

Table 1 shows our submissions obtained consistent results for the development and test sets. The SOLAR model (Kim et al., 2023) outperformed Orca2 (Mitra et al., 2023), Quwen (Bai et al., 2023) and Mistral (Jiang et al., 2023) LLMs for all metrics.

Our SOLAR LLM achieved a control F1 score of 0.7790, significantly outperforming our submission for the NLI4CT-2023 task (F1 of 0.7091) based on ensemble BioLinkBERT transformers (Chen et al., 2023). This confirms that using LLMs properly can outperform pre-trained language models for the same task. In addition, the number of parameters in the LLM doesn't directly influence performance, indicating that model architecture is more important rather than scale.

In our proposed system workflow, regardless of which LLM model was used as the main framework for the NLI task, a higher faithfulness score was achieved when compared with the consistency score. This indicates that an LLM usually makes correct predictions with correct reasons.

In summary, in the NLI4CT-2024 task, our system based on the SOLAR model produced a promising faithfulness score of 0.9236, ranking second place among 32 participating systems, and ranked first for consistency with a score of 0.8092.

4 Conclusions

This study describes the NYCU-NLP submission for the SemEval-2024 NLI4CT task, including system design, implementation and evaluation. We aggregated several LLMs to determine the inference relation between CTRs and statements that may be manipulated with designed interventions to investigate the faithfulness and consistency of the developed models. Our system obtained a faithfulness score of 0.9236, ranking second among all 32 participating teams, and ranked first for consistency with a score of 0.8092.

[1] https://huggingface.co/upstage/SOLAR-10.7B-Instruct-v1.0
https://huggingface.co/microsoft/Orca-2-13b
https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca
https://huggingface.co/Qwen/Qwen-14B-Chat

Acknowledgments

This study is partially supported by the National Science and Technology Council, Taiwan, under the grant NSTC 111-2628-E-A49-029-MY3.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fanm Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Lin, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv:2309.16609v1*. https://doi.org/10.48550/arXiv.2309.16609

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: a pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, pages 3615-3620. http://dx.doi.org/10.18653/v1/D19-1371

Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. 2019. Overview of the MEDIQA 2019 Shared Task on Textual Inference, Question Entailment and Question Answering. In *Proceedings of the 18th Biomedical Natural Language Processing Workshop and Shared Task*, Association for Computational Linguistics, pages 370–379. http://dx.doi.org/10.18653/v1/W19-5039

Chao-Yi Chen, Kao-Yuan Tien, Yuan-Hao Cheng, and Lung-Hao Lee. 2023. NCUEE-NLP at SemEval-2023 Task 7: Ensemble biomedical LinkBERT transformers in multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 776-781. https://doi.org/10.18653/v1/2023.semeval-1.107

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yangping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quo V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv:2210.11416v5*. https://doi.org/10.48550/arXiv.2210.11416

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient attention with IO-awareness. *arXiv:2205.14135v2*. https://doi.org/10.48550/arXiv.2205.14135

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, pages 4171-4186. http://dx.doi.org/10.18653/v1/N19-1423

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *Proceedings of the 11th International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv:2106.09685v2*. https://doi.org/10.48550/arXiv.2106.09685

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2020. ClinicalBERT: Modeling clinical notes and predicting hospital readmission. *arXiv: 1904.05342v3*. https://doi.org/10.48550/arXiv.1904.05342

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lelio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothee, Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv: 2310.06825v1*. https://doi.org/10.48550/arXiv.2310.06825

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 Task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Donald Landers, and André Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023*

Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pages 16745-16764. https://doi.org/10.18653/v1/2023.emnlp-main.1041

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Donald Landers, and André Freitas. 2023b. SemEval-2023 Task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 2216-2226. https://doi.org/10.18653/v1/2023.semeval-1.307

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv: 2305.18290v2*. https://doi.org/10.48550/arXiv.2305.18290

Kamal Raj Kanakarajan, and Malaikannan Sankarasubbu. 2023. Saama AI research at SemEval-2023 Task 7: Exploring the capabilities of Flan-T5 for multi-evidence natural language inference in clinical trail data. In *Proceedings of the 17th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 995-1003. https://doi.org/10.18653/v1/2023.semeval-1.137

Dahyun Kim, Chanjun Park, Sanghoom Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjim Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2023. SOLAR 10.7B: Scaling large language models with simple yet effective depth up-scaling. *arXiv:2312.15166v2*. https://doi.org/10.48550/arXiv.2312.15166

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4): 1234-1240. https://doi.org/10.1093/bioinformatics/btz682

Lung-Hao Lee, Yi Lu, Po-Han Chen, Po-Lei Lee, and Kou-Kai Shyu. 2019. NCUEE at MEDIQA 2019: medical text inference using ensemble BERT-BiLSTM-Attention model. In *Proceedings of the 18th Biomedical Natural Language Processing Workshop and Shared Task*. Association for Computational Linguistics, pages 528-532. http://dx.doi.org/10.18653/v1/W19-5058

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 4487-4496. https://doi.org/10.18653/v1/P19-1441

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi. Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason. *arXiv:2311.11045v2*. https://doi.org/10.48550/arXiv.2311.11045

OpenAI. 2023. ChatGPT (Large language model). https://chat.openai.com/chat

Alexey Romanov, and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1586-1596. https://www.aclweb.org/anthology/D18-1187

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023a. OpenChat: Advancing open-source language models with mixed-quality data. *arXiv:2309.11235v1*. https://doi.org/10.48550/arXiv.2309.11235

Weiqi Wang, Baixuan Xu, Tianqing Fang, Lirong Zhang, and Yangqiu Song. 2023b. KnowComp at SemEval-2023 Task 7: Fine-tuning pre-trained language models for clinical trial entailment identification. In *Proceedings of the 17th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 1-9. https://doi.org/10.18653/v1/2023.semeval-1.1

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *Proceedings of the 10th International Conference on Learning Representations*. https://openreview.net/forum?id=gEZrGCozdqR

Yichong Xu, Xiaodong Liu, Chunyuan Li, Hoifung Poon and Jianfeng Gao. 2019. DoubleTransfer at MEDIQA 2019: Multi-source transfer learning for natural language understanding in the medical domain. In *Proceedings of the 18th Biomedical Natural Language Processing Workshop and Shared Task*. Association for Computational Linguistics, pages 399-405. https://doi.org/10.18653/v1/W19-5042

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association

for Computational Linguistics, pages 8003-8016. http://dx.doi.org/10.18653/v1/2022.acl-long.551

Sicheng Yu, Jing Jiang, Hao Zhang, Yulei Niu, Qianru Sun, and Lidong Bing. 2022. Interventional training for out-of-distribution natural language understanding. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11627-11638. https://doi.org/10.18653/v1/2022.emnlp-main.799

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. THiFLY research at SemEval-2023 Task 7: A multi-granularity system for CTR-based textual entailment and evidence retrieval. In *Proceedings of the 17th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 1681-1690. https://doi.org/10.18653/v1/2023.semeval-1.234

Team MLab at SemEval-2024 Task 8: Analyzing Encoder Embeddings for Detecting LLM-generated Text

Kevin Li†
Stanford University
kevinli7@stanford.edu
Kenan Hasanaliyev†
Stanford University
kenanhas@stanford.edu
Sally Zhu
Stanford University
salzhu@stanford.edu
George Altshuler
Stanford University
gwa@stanford.edu
Alden Eberts
Stanford University
ajeberts@stanford.edu
Eric Chen
Stanford Univeristy
ericc27@stanford.edu
Kate Wang
Stanford University
kyw1923@stanford.edu
Emily Xia
Stanford University
emxia18@stanford.edu
Eli Browne
Stanford University
ebrowne@stanford.edu
Ian Chen
Stanford University
ianychen@stanford.edu
Umut Eren
Stanford University
umuteren@stanford.edu

Abstract

This paper explores solutions to the challenges posed by the widespread use of LLMs, particularly in the context of identifying human-written versus machine-generated text. Focusing on Subtask B of SemEval 2024 Task 8, we compare the performance of RoBERTa and DeBERTa models. Subtask B involved identifying not only human or machine text but also the specific LLM responsible for generating text, where our DeBERTa model outperformed the RoBERTa baseline by over 10% in leaderboard accuracy. The results highlight the rapidly growing capabilities of LLMs and importance of keeping up with the latest advancements. Additionally, our paper presents visualizations using PCA and t-SNE that showcase the DeBERTa model's ability to cluster different LLM outputs effectively. These findings contribute to understanding and improving AI methods for detecting machine-generated text, allowing us to build more robust and traceable AI systems in the language ecosystem.

1 Introduction

We live in a society that currently relies heavily on the use of LLMs (Large Language Models), which has followed from the explosive popularity of ChatGPT when it was released in late 2022. Now, with the introduction of GPT-4 and other more powerful LLMs, it has become increasingly important for us to have the ability to distinguish human-written text from machine-generated text. The fluency of recent models, paired with their tendency to hallucinate, has given rise to a very natural concern that there could be both accidental and intentional "bad actors" seeking to spread false information. Research has indicated that about one in every five jobs has over half of its tasks incorporated into LLMs, and that statistic is positively correlated with the barrier to entry (Eloundou et al., 2023). Consequently, these models have the necessary training data to spit out an immense number of plausibly correct but actually incorrect texts, which would be extremely detrimental because humans historically have been unable to distinguish between them beyond the level of random guessing. Such results are supported with recent work aiming to distinguish human-written sentences from AI-generated ones, with the AuTexTification study additionally demonstrating that cross-domain AI-generated text detection from Bloomz or GPT is more difficult in non-English languages (Sarvazyan et al., 2023). Furthermore, in efforts to facilitate unbiased dataset generation for related studies, the framework of TextMachina was created, and it contains crucial post-processing abilities like removing disclosure patterns and truncation (Sarvazyan et al., 2024). SemEval-2024's Task 8 (Wang et al., 2024) attempts to provide a working solution

† Indicates equal contribution amongst authors

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1463–1467

to the above problems by using standalone, self-operational means to classify whether a given text was authentically written by a human or artificially generated by a machine, in hopes of eventually building toward a foolproof method of detecting misinformation.

In subtask A, we were tasked with creating a binary classification model to determine if a given text was human-written or machine-generated. Within this subtask, there are two different tracks: one for monolingual (only English) and another for multilingual sources (something about which track we did or if we ended up doing both). Out of curiosity, we used base DeBERTa with default hyperparameters as our submission for this subtask, but it only performed 5% worse than the RoBERTa baseline. This specific subtask is important because LLMs are becoming more powerful and easily accessible, so there is a larger potential for misuse. This classification task would help catch the people who are misusing this technology to harm society.

In Subtask B, we trained a neural network to identify not only whether a given text was human-written or machine-written, but also to identify which large language model was responsible for generating that text. These language models include ChatGPT, Cohere, Dolly, and more. This task is important for many of the same reasons as subtask A; as LLMs are becoming more capable and accessible, being able to distinguish between human and model is crucial. Furthermore, being able to distinguish between different models allows for better enforcement of AI-safety laws and accountability.

2 Methods

2.1 RoBERTa

The baseline performances provided for both subtasks A and B revolved around HuggingFace's RoBERTa model. RoBERTA was developed to enhance the usability of post-BERT models, and incorporated a variety of techniques including longer training times, larger batches, more data, the elimination of the next sentence prediction objective, longer training sequences, and dynamic modification of the masking pattern (Liu et al., 2019). For the monolingual component of subtask A, roberta-base was used for a baseline of about 0.88, and for the multilingual component, xlm-roberta was used (to account for the various other languages) for a baseline of about 0.81. For subtask B, roberta-base was used again for a baseline of about 0.75.

2.2 DeBERTa

DeBERTa was designed as an upgrade to BERT and RoBERTa with the addition of disentangled attention and an enhanced mask decoder, and furthermore, its fine-tuning included adversarial training (He et al., 2021). As a result, we used the deberta-base model from HuggingFace with the assumption that it would outperform RoBERTa, and our best model did. Regarding hyperparameter tuning, we wanted to fine-tune the deberta-base model on the given dataset, so we looped through learning rates of 1e-5, 5e-5, and 1e-4, batch sizes of 4 and 8, epoch counts of 2 and 3, and weight decay constants of 1e-3, 5e-3, and 1e-2. By truncating the input length to a constant 1024 tokens, we established that the optimal hyperparameters (at least from what we tested) that yielded the highest accuracy were a learning rate of 1e-5, a batch size of 4, an epoch count of 3, and a weight decay constant of 1e-2.

2.3 Model interpretibility

To further analyze the inner working of the DeBERTa model, we analyzed our trained model's pooled outputs that encode the input sentence as whole prior to the logits. We used two dimensionality reductions algorithms, namely PCA and t-SNE for 2-D projection. PCA operates by finding orthogonal directions with the highest variance and projecting to the subspace spanned by the orthogonal directions. The t-SNE algorithm (van der Maaten and Hinton, 2008) works by preserving pairwise similarities in the data to generate related clusters. Our t-SNE projections were computed with a perplexity of 35 and iteration count of 300. Both algorithms were run on the 18,000 sentences in the test data for subtask B.

3 Results

The final DeBERTa model had the following results on the validation set, with a weighted average of 0.98633 precision, 0.98599 recall, and 0.98599 F1-score.

In comparison, our RoBERTa model performed worse, with each F1-score being lower than the corresponding F1-score for the DeBERTa model.

This model had weighted scores of 0.97979 precision, 0.97909 recall, and 0.97909 F1-score. Although these were high, they were each still a

Label	Source	Precision	Recall	F1-Score
0	Human	0.99916	0.99375	0.99645
1	ChatGPT	0.94944	0.99417	0.97129
2	Cohere	0.98735	0.99824	0.99276
3	Davinci	0.98912	0.94708	0.96765
4	Bloomz	1.0	0.99833	0.99917
5	Dolly	0.99311	0.98610	0.98606

Table 1: DeBERTa results on the Subtask B validation set

Label	Source	Precision	Recall	F1-Score
0	Human	0.99916	0.98792	0.99351
1	ChatGPT	0.93008	0.99250	0.96027
2	Cohere	0.97631	1.0	0.98801
3	Davinci	0.98367	0.92875	0.95542
4	Bloomz	0.99791	0.99667	0.99729
5	Dolly	0.99170	0.96966	0.98055

Table 2: RoBERTa results on the Subtask B validation set

bit lower than the corresponding metrics for the DeBERTa model.

We compared our model's predictions for the test set with the labels provided. The total accuracy was 0.8266666667. Below is the confusion matrix in table form for the predicted labels versus actual labels, with the labels corresponding to their respective sources. For example, the 439 entry has predicted label 2 and actual label 3, meaning that there were 439 predictions for the Cohere source that were actually from the Davinci source.

	Actual					
Pred	0	1	2	3	4	5
0	2050	3	9	237	541	151
1	0	2823	6	171	0	0
2	11	208	2342	439	0	11
3	27	624	3	2334	12	27
4	0	1	0	1	2997	1
5	77	187	34	626	541	4612

Table 3: Confusion matrix for Subtask B labels

The results on the test set are summarized in the table below.

Additionally, we analyzed the pooled outputs of the trained DeBERTa model on subtask B's data. To visualize the outputs in 2-D, PCA and t-SNE projection techniques were applied on the 768-D pooled outputs. The data points were colored by their corresponding text source (either human or LLM model) in Figures 1 and 2 on the test set.

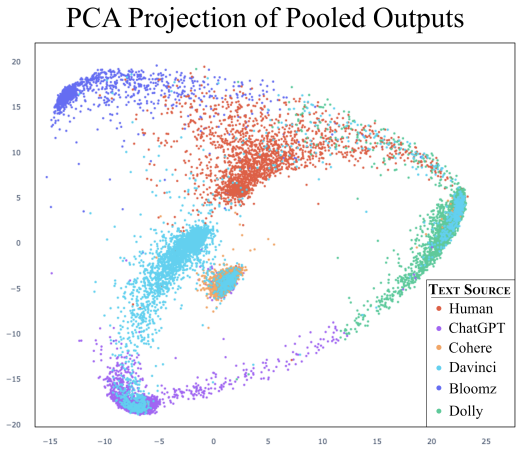


Figure 1: PCA projection of pooled outputs on the subtask B test data

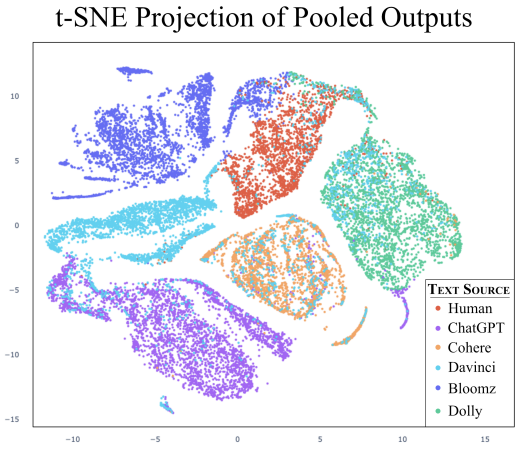


Figure 2: t-SNE projection of pooled outputs on the subtask B test data

Label	Source	Precision	Recall	F1-Score
0	Human	0.95	0.69	0.80
1	ChatGPT	0.73	0.94	0.82
2	Cohere	0.98	0.78	0.87
3	Davinci	0.65	0.78	0.71
4	Bloomz	0.84	1.00	0.91
5	Dolly	0.93	0.78	0.85

Table 4: DeBERTa test results on Subtask B

4 Discussion

4.1 Validation and Test Set Results

For both the RoBERTa statistics and the DeBERTa statistics in their validation results, the top two entries for precision and F1-score were Bloomz and Human in some order, which indicates a greater degree of identifiability from these sources. Bloomz was trained on multilingual tasks and fine-tuned on English prompts (Muennighoff et al., 2023), and many online texts are written by humans who are at least bilingual, so there could be a mannerism of text generation tied to multilingualism that makes it easier to pinpoint and distinguish these sources from the others.

Another observation is that Davinci performed reasonably well across the board for the validation sets, but had abysmal scores for the DeBERTa test set. Considering that DeBERTa is a more recent model, it is entirely possible that it has more difficulty with older data, which may explain why Davinci did as poorly as it did. However, besides Davinci and Bloomz being outliers on either end of the spectrum for F1-score, the rest of the values fell within a generally stable range, indicating that DeBERTa had a balanced evaluation of texts.

Additionally, interestingly enough, ChatGPT ranks last or near last in precision and F1-score in all the tables, but makes up for that with its high recall values. This could mean that the text was detected to be AI-generated with relative ease, but was then often misclassified as being from another AI source. Given that GPT-4 has greatly enhanced abilities compared to its predecessor, swapping out ChatGPT for GPT-4 could yield radically different results (for a potential future direction).

4.2 Visualizations of Pooled Outputs

It is clear from both the PCA and t-SNE visualizations that the DeBERTa model is successfully able to distinguish between different LLMs and human output in distinct clusters. Of note, however, are the blue points corresponding to Davinci text located in clusters of different colors. This phenomenon follows from recent research and shows that human writing tasks can still be quite susceptible to LLM influence due to their positive association with exposure (Eloundou et al., 2023). We speculate that Davinci being one of the earliest models influenced the training data of the other models that came on later, causing them to write similarly to Davinci. This supports our earlier hypothesis from the raw results, but seemingly contradicts findings that Davinci exhibits fewer confusions and is thus easily distinguishable from other models (Sarvazyan et al., 2023). One possible explanation for this is that our visualizations used parameters that clearly confined the other sources to their regions; it is entirely possible that a different configuration of parameters would yield a graph that displays an obvious Davinci scatter area while having a jumble of colors elsewhere for the other models.

5 Conclusion

For SemEval 2024 Task 8, Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection, Team MLab submitted models for subtasks A and B. Specifically for subtask B, we used a base DeBERTa model and significantly outperformed the provided baseline set by a base RoBERTa model, with our model's final accuracy coming out to 0.827. In analyzing the precision, recall, and F1-score statistics, we discovered trends in the recorded values that seem to indicate that the method of training models, as well as the timeline of their training, have profound effects on the detectability of machine-generated text. Finally, by creating and interpreting PCA and t-SNE graphs, we present visual evidence that DeBERTa's internal reasoning groups various LLM results in separate clusters, even though Davinci acted as an exception with its colored points scattered in the general vicinity of other models. Therefore, visualizing AI thought processes can provide us with useful insights regarding how we can understand and improve the language ecosystem that they share with us.

References

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. Gpts are gpts: An early look at the labor market impact potential of large language models.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Areg Mikael Sarvazyan, José Ángel González, and Marc Franco-Salvador. 2024. Textmachina: Seamless generation of machine-generated text datasets.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Calc-CMU at SemEval-2024 Task 7: *Pre-Calc* - Learning to Use the Calculator Improves Numeracy in Language Models

Vishruth Veerendranath and **Vishwa Shah** and **Kshitish Ghate**
Carnegie Mellon University
{vveerend, vishwavs, kghate}@andrew.cmu.edu

Abstract

Quantitative and numerical comprehension in language is an important task in many fields like education and finance, but still remains a challenging task for language models. While tool and calculator usage has shown to be helpful to improve mathematical reasoning in large pretrained decoder-only language models, this remains unexplored for smaller language models with encoders. In this paper, we propose **Pre-Calc**, a simple pre-finetuning objective of learning to use the calculator for both encoder-only and encoder-decoder architectures, formulated as a discriminative and generative task respectively. We pre-train BERT and RoBERTa for discriminative calculator use and Flan-T5 for generative calculator use on the MAWPS, SVAMP, and AsDiv-A datasets, which improves performance on downstream tasks that require numerical understanding. Our code and data are available at https://github.com/calc-cmu/pre-calc.

1 Introduction

The advancement of language modeling in natural language processing has significantly impacted various computational tasks. However, the intricacy of numerical and quantitative comprehension in text remains a challenging frontier. Numerals, unlike words, possess unique characteristics that necessitate specialized handling by language models either in tokenization or processing. This necessity becomes particularly evident in tasks involving quantitative reasoning, where the ability to interpret and manipulate numerical information is crucial.

Numeracy involves majorly 2 properties. The first is *semantic reasoning* which focuses more on the understanding of relations in text and the second is *computational abilility* which focuses on performing explicit mathematical operations. Hence, the aim is to develop systems that can perform explicit mathematical operations while retaining or improving its quantitative reasoning.

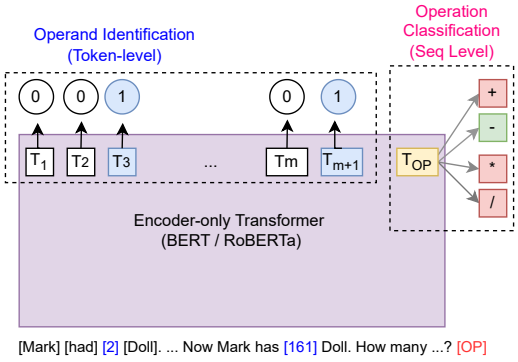


Figure 1: Pre-Calc for Encoder-Only models

We present **Pre-Calc**, a pre-finetuning objective of learning to use the calculator, to improve numerical abilities in language models. We propose Pre-Calc objectives for both the encoder-only and encoder-decoder classes of language models, and use a combination of the MAWPS (Koncel-Kedziorski et al., 2016), SVAMP (Patel et al., 2021), AsDiv-A (Miao et al., 2020) datasets to pre-finetune the models.

Our encoder-only objective, used to pre-finetune BERT and RoBERTa models, offers quick and efficient processing suitable for tasks where speed is paramount. The Pre-Calc versions of the models show competent performance on all quantitative downstream tasks from NumEval (Chen et al., 2023) and substantial improvements on 4 out of 6 sub-tasks with an improvement greater than 10 points for RedditNLI and AWPNLI specifically.

Similarly, our encoder-decoder approach, used to pre-finetune Flan-T5, demonstrates an improved ability to perform explicit computations in computation-intensive tasks like AWPNLI. Although there is a noted trade-off, with a slight decrease in performance on text-focused and semantic tasks, the objective showcases strengths in processing mathematically intensive language.

Our study underscores the potential of tailored

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1468–1475

language models to significantly enhance numeracy in NLP, providing an avenue for more efficient and effective processing of numerical data in language.

2 Task and Data

2.1 Downstream Tasks

We focus on the QNLI and QQA tasks of NumEval Task 1 (Chen et al., 2023) as our downstream tasks.

QNLI is the task of making *natural language inferences* based on quantitative clues. This dataset is adopted from the EQUATE(Ravichander et al., 2019) and is composed of NewsNLI, RedditNLI, AWPNLI and RTE-Quant. StressTest involves numerical reasoning instances from Naik et al. (2018), used as a synthetic sanity check.

QQA involves the task of *multiple-choice question answering* involving commonsense as well as quantitative comparisons. The dataset for this is adopted from Task 3 of NumGLUE (Mishra et al., 2022) and the Quarel dataset (Tafjord et al., 2019), which includes questions from quantitative domains such as physics and economics.

2.2 Pre-Finetuning Data

We use the MAWPS dataset (Koncel-Kedziorski et al., 2016), SVAMP (Patel et al., 2021) and AsDiv-A (Miao et al., 2020) as the numerical domain datasets for pre-finetuning. These consist of simple arithmetic word problems, along with their numerical solutions. The three datasets create a dataset with **4,225** total examples which are challenging and require understanding the context of numbers, represented either as digits or in words.

We construct this dataset from the Calc-X collection (Kadlčík et al., 2023) that has been annotated with equations for each problem, as well as `<gadget>` annotations in the answer to train a model to use a calculator when the `<gadget>` token is produced. We use the annotations of the equations particularly in our methodology.

3 Pre-Calc Methodology

We posit that learning to use a calculator requires understanding of numbers and ways in which numbers can be combined. This is used to formulate the **Pre-Calc** objectives described below.

3.1 Encoder-Only

3.1.1 Data Preprocessing

We preprocess Calc-MAWPS, Calc-SVAMP and Calc-AsDiv-A (from the Calc-X collection) (Kadlčík et al., 2023) and add 2 new features required for Pre-Calc. First is the *operand tag sequence*, which is a sequence of binary tags that is 1 if the original token it corresponds to is an *operand* and 0 if it isn't. Secondly we extract the *Operation*, which is the operation among {+ (add), - (subtract), * (multiply), / (divide)} that is required for the question. We extract the operation either directly from the equation or the reasoning chain in Calc-X and generate the operand tag sequence, by first extracting the operands and then tagging the occurances of the operands in the binary sequence with a 1. As part of this process we also filter out instances where there are more than one distinct operations as part of the equation.

3.1.2 Pre-Calc Method

An illustration of the Pre-Calc method for Encoder-only model can be seen in Fig 1. This is decomposed into two tasks as a dual-objective.

Firstly, we use the pretrained Encoder-only language model for the task of *Operand Identification*, which is a token-level classification task. The tags possible for each token are 1 and 0.

Secondly, we perform the task of *Operation Classficiation* by adding a special `[OP]` token at the end of each sequence and using this `[OP]` token's final layer representation to classify the operation required in this sequence (+, -, *, /). Hence, this is essentially a sequence-level classification task similar to classifying from the representation of a `[CLS]` token. However, we do not use the `[CLS]` token at the start of the sequence, to enable this objective even in non-bidirectional models with an autoregressive attention mask (like decoder-only models).

In essence, we use two heads — one token classification head for Operand Identification, and one sequence classification head for Operation Classification — to train it with the dual objective as per Equation 1

$$\mathcal{L} = \mathcal{L}_{operation} + \lambda \mathcal{L}_{operand} \quad (1)$$

where $\mathcal{L}_{operation}$ is the cross-entropy loss for the sequence classification (`[OP]`) head and $\mathcal{L}_{operand}$ is the binary cross entropy (BCE) loss for the token classification head. Here we empirically set $\lambda = 1$.

3.1.3 Downstream Task Inference

For most downstream tasks, we do not explicitly perform calculator computations using operands and operations predicted by the model and instead

use Pre-Calc only as a learning objective before finetuning it for specific downstream tasks. However, as AWPNLI task requires the model to be able to perform calculations explicitly, we utilize an alternative strategy for its inference adapted from our pre-finetuning strategy shown in Fig 1. We first extract the operand labels for each token from the premise T_i and operation using the T_{op} token. This gives us the operands and operation, after which we automate the calculation of the final answer comparing it with the hypothesis. This helps the model focus on the semantic extraction of operation and offload explicit computation to the calculator.

3.2 Encoder-Decoder

Encoder-Decoder or Decoder based models provide the abilities of long-form unbounded generations. This is advantageous for numerical problems, where multiple intermediate operations might be required for computation or reasoning(Wei et al., 2022). By reframing our task to output expressions, we distil the task to output the set of operations, leaving the computation part to the tool.

3.2.1 Data Preprocessing

As mentioned earlier, we use the MAWPS dataset for training our model to output expressions. As each instance in MAWPS consists of a question and a single numerical answer, to obtain closer resemblance to NLI format tasks, we reframe the question-numerical answer instance to a pair of complete sentences using prompting with LLaMa-7B (Touvron et al., 2023) as this is a simple text generation task. To obtain contradiction pairs, we perturb the true numerical answer by a small value (ranging from -5 to +5) before passing to the LLaMa-7B model as these will create harder instances for the model to learn from. We additionally also use the Multi-NLI(Williams et al., 2018) dataset to retain and improve textual inferential abilities of the model. We train on this combined data as Seq2Seq generation task. Combining these tasks should allow the model to infer both semantic and computational capabilities.

3.2.2 Pre-Calc Method

We utilize this ability of Seq2Seq modeling by finetuning Flan-T5 on our NLI-based tasks for pre-finetuning mentioned in 2.2. As shown in Figure 2, we use a **`math-nli`** prefix tag for tasks that require mathematical computation eg: MAWPS reformatted as above and use a **`text-nli`** tag for text-based

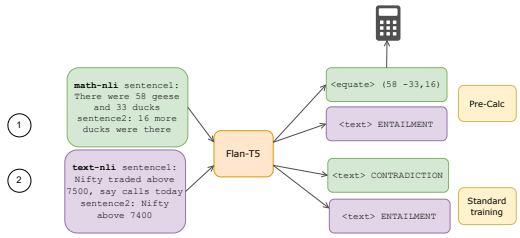


Figure 2: Our Encoder-Decoder based approach

tasks eg: Multi-NLI. This lets the model decipher whether the task requires explicit calculation - in which case it should output an expression for tool use, or use its inherent text-capabilities to reason over text numeracy.

Similar to Kadlčík et al. (2023), we make the model output token **`<equate>`** with the corresponding expression for computational tasks as essentially the task involves equating expressions in sentence 1 and sentence 2 and **`<text>`** for more textual-numeracy tasks with the final answer. This helps at inference to verify if the final computation requires to go through the calculator or not. We also hope to expand to **`<compare>,<compute>`** tokens as we extend this method in future to more down-stream tasks. We denote this as our Pre-Calc method for Encoder-Decoder models which performs tool-based pre-finetuning. As our baseline we also evaluate the performance of doing only text-based fine-tuning which we call our Standard Training approach.

4 Experiments

4.1 Baselines

We compare the performance of our method against several baseline models on tasks that require numeracy. Following Chen et al. (2023), the baseline methods involve reframing techniques, namely *Original*, *Digit-based*, and *Scientific* Notation methods, and are pre-finetuned on the *Comparing Numbers Dataset (CND)*. Each of these methods are applied to both BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) to create two versions of each baseline method.

4.2 Encoder-Only

We use the pretrained BERT and RoBERTa base models and pre-finetune as per Section 3.1.2. We use the 4-class cross-entropy loss for training the *Operation Classification* head, and a 2-class cross-entropy (equivalent to binary cross entropy) loss for the *Operand Identification* head. The models

Model	Notation	QNLI					QQA
		RTE-QUANT	News	Reddit	AWPNLI	Streess Test	
BERT	Org	66.73	74.22	62.40	59.20	99.46	**56.79**
	Digit	60.22	75.94	62.86	53.20	**99.70**	52.63
	Sci	66.80	75.60	65.14	60.73	99.46	53.33
	CN-Digit	62.88	76.97	68.57	60.27	99.58	53.60
	CN-Sci	66.87	**77.98**	65.64	54.70	99.58	52.38
	Pre-Calc (ours)	**67.00**	76.54	**76.00**	**68.97**	99.47	53.93
RoBERTa	Org	62.79	78.35	59.33	57.64	**100.00**	52.27
	Digit	62.67	79.38	63.71	56.69	99.94	58.94
	Sci	62.93	79.37	62.88	57.41	**100.00**	56.47
	CN-Digit	68.13	77.66	62.99	**58.80**	**100.00**	51.21
	CN-Sci	63.97	74.57	63.80	58.74	99.98	53.6
	Pre-Calc (ours)	**73.90**	**82.21**	**78.00**	58.17	**100.00**	**61.05**

Table 1: Micro-F1-Scores (in %) of Pre-Calc trained models as compared to CN (Comparing Numbers) trained and reframing (Digit, Sci) baselines

are trained with the Adam optimizer for 20 epochs, a batch size of 8, and a learning rate of 5e-4. The checkpoint after this pre-finetuning is named Pre-Calc-BERT[1] or Pre-Calc-RoBERTa[1].

We then finetune Pre-Calc-BERT and Pre-Calc-RoBERTa on the downstream tasks of QNLI and QQA using the same hyperparameters used by the CN-BERT baselines (Chen et al., 2023) — AdamW optimizer with a learning rate of 5e-5, batch size of 8 for 5 epochs. We use 10-fold cross validation to report our results for the tasks where an explicit test split is not available.

4.3 Encoder-Decoder

We use Flan-T5 as our base model, For pre-finetuning, we collect a balanced sample consisting about 4200 instances created from MAWPS for `math-nli` task and 3900 instances extracted from Multi-NLI for `text-nli` task.[1]

As this is a sequence generation task, the objective is same as that in CausalLM modeling for next-word prediction. We use AdamW optimizer with a learning rate of 5e-5, batch size of 8 for 5 epochs. We do not perform any fine-tuning on our downstream tasks, and show results for prompt based few-shot evaluations on each task. We call our model FlanT5-Pre-Calc[1] and use 2 baselines, Flan-T5 few-shot and Flan-T5-ST only with standard text training[1].

[1] https://huggingface.co/collections/Calc-CMU/pre-calc-657a5ad5f1ae42fb12364563

5 Results and Discussion

5.1 Encoder-Only

Our evaluation on the QNLI and QQA tasks, as outlined in Table 1, demonstrates the efficacy of our Pre-Calc approach. For BERT, our Pre-Calc method significantly outperforms all other reframing techniques for RedditNLI, AWPNLI and RTE-Quant. These results highlight the effectiveness of our method in dealing with diverse numerical information in natural language. In the case of RoBERTa, the Pre-Calc approach consistently outperformed other methods across all three tasks - RTE-Quant, NewsNLI and RedditNLI. This performance is markedly superior compared to the original RoBERTa and other variants that use the reframing techniques, with lower scores in all categories.

For AWPNLI we report results for baselines from Chen et al. (2023), and for our results we compute F1-score on the complete dataset using our methodology described in section 4.2. We see a substantial improvement in Pre-Calc compared to the earlier baselines which can be attributed to our training and inference strategy which can precisely attend and compute an expression which is essential for the AWPNLI task.

In QQA as well, Pre-Calc-RoBERTa improves performance over its counterpart. This indicates that Pre-Calc improves commonsense reasoning abilities and this effect is more pronounced in RoBERTa which is a stronger base model.

Overall, the results validate our hypothesis that

Model \Task	AWP NLI	News NLI	RTE Quant
Few-shot	41.56	77.47	**85.74**
ST (ours)	37.55	**76.75**	73.43
Pre-Calc (ours)	**80.29**	75.20	71.26

Table 2: Micro F1-score of Flan-T5-large when using our Encoder-Decoder based approach

the Pre-Calc approach, which integrates calculator-like capabilities into the model, significantly enhances performance in tasks requiring numeral-aware semantic and computational capabilities .

5.2 Encoder-Decoder

We present the results for our Encoder-Decoder based approach in Table 2. We see that for AWPNLI which requires explicit computation, FlanT5(Pre-Calc) achives almost double performance compared to FlanT5-few shot and FlanT5-ST, showing that Flan-T5 originally did not have this capability to evaluate expressions and compare values and this property cannot be instilled only via text finetuning as can be seen from the performance of FlanT5-ST. Further compared to prior works (Chen et al., 2023), this achieves state-of-the art results on AWPNLI.

However, we see that the performance slightly decreases on NewsNLI and RTEQuant which are more text-focused tasks. We see that original pre-trained FlanT5 does better at this as it already has inherent properties to handle semantic numeracy. This is likely because training with specific tasks discussed above causes forgetting/over-fitting in the model. This can also be attributed to the language-modeling MLE loss which focuses more on generating outputs specific to the format discussed in Fig 2 rather than its original properties of in-context learning and reasoning. To combat this, in the future we hope to regularize learning better so that a diversity of tasks can be included avoiding overfitting in the model.

6 Analysis

6.1 Dual-Objective in Encoder-Only Pre-Calc

We inspect the characteristics of the two objectives during pre-finetuning. Fig. 3 shows the F1-score across epochs for the operand identification objective on the validation data. While this seems to fluctuate, it consistently stays above 90% (the accuracy for this task also consistently remains at about 99%), indicating that the operand identification task is not very challenging and that there is very little loss signal from this task beyond the first few epochs. Regardless of having the second objective, the F-1 for this task is still maintained at a high number.

In Fig. 4 we see the accuracy plot on validation data for the operation classification objective across epochs. Here we see that the accuracy consistently increases but still remains under 75% which tells us that this objective is a lot more challenging, which is also explained by the fact that it has to be inferred from text. Together, the two aid different abilities — picking numbers out with operand identification and combining numbers with operation classification — which are both important for any downstream quantitative task

6.2 Operation wise difficulty for FlanT5

We sample 500 instances from the MAWPS reframed dataset, to observe operation-wise accuracy for the model. We observe in Figure 5 that about 60% errors are for instances that entail a divide operation. This could be because understanding division requires the model to develop an understanding of what operand should be the numerator and which should be the denominator. There are also rare instances where the model is required to understand the idea of ratio-proportion which requires more complex understanding compared to other operations.

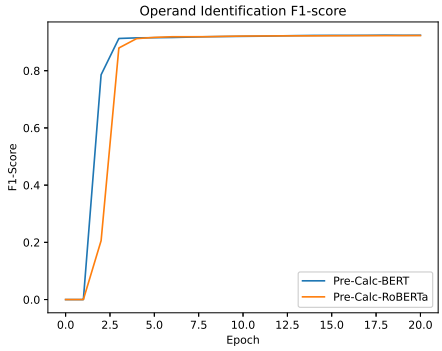


Figure 3: Operand Identification Loss Plot

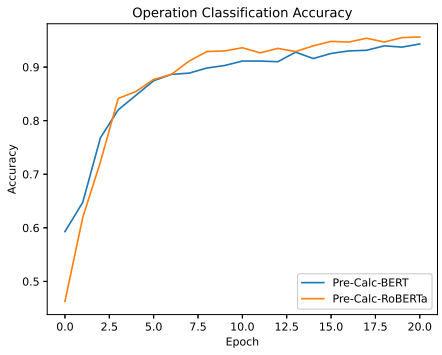


Figure 4: Operation Classification Loss Plot

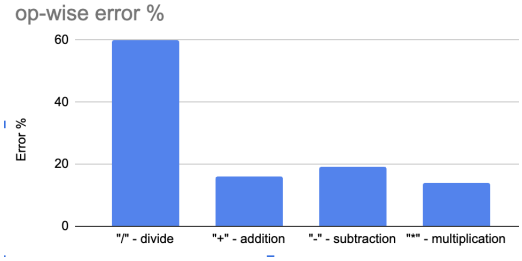


Figure 5: Operation wise error for FlanT5-Pre-Calc

7 Related Work

Numeracy in LMs Numeracy, or the ability to understand and work with numbers, is a critical aspect that has been relatively underexplored compared to other linguistic competencies in NLP models. Spithourakis and Riedel (2018) emphasized the need for LMs to better understand numbers, setting a precedent for subsequent research.

Chen et al. (2019) introduced Numeracy-600K, a large-scale dataset designed to improve the ability of models to detect exaggerated information in financial texts. Concurrently, Wallace et al. (2019) explored the embedding properties of numbers, shedding light on how numeracy can be integrated into LMs. Zhang et al. (2020) analyzed the representation of numerals in scientific notation, addressing the challenge of scale understanding in LMs. Chen et al. (2021) furthered this exploration by suggesting a digit-based encoder for numeral encoding, providing a novel perspective on numeral representation.

Pre-Finetuning In addition to these studies focused on numeral representation, other researchers have investigated the potential of pre-finetuning tasks to enhance LM capabilities. Aghajanyan et al. (2021) introduced a massive multi-task representation with pre-finetuning, demonstrating the efficacy of pre-finetuning in improving model performance across a range of tasks.

Geva et al. (2020) proposed GENBERT, which is trained on automatically-generated synthetic data in a multi-task setup. This training significantly improves performance on numerical reasoning tasks such as DROP and math word problems, while maintaining high performance on standard reading comprehension tasks. Wang et al. (2017) presented a deep neural solver, a hybrid model combining the RNN with a similarity-based retrieval to translate math word problems into equation templates.

Tool-Use Gou et al. (2023) presented a series of Tool-integrated Reasoning Agents (ToRA) designed to solve complex mathematical problems by augmenting the model with external computational tools. The training process involves collecting interactive tool-use trajectories and applying imitation learning and output space shaping, showcasing the efficacy of combining natural language reasoning with program-based tool use. Kadlčík et al. (2023) introduced Calc-X, a collection of datasets designed to integrate calculator usage into language model reasoning chains. Calc-X consolidates 300,000 samples from several chain-of-thought tasks requiring arithmetic reasoning. The study demonstrates how Calcformers, models trained on Calc-X, significantly enhance the accuracy of generating correct results by offloading computations to symbolic systems.

8 Conclusion and Future Work

In this work, we improve the numeracy in language models on the QNLI and QQA tasks which involve textual and computational quantitative reasoning. We do so by proposing calculator usage as a pre-finetuning task in a discriminative and generative fashion for encoder-only and encoder-decoder models respectively. This improves encoder-only models across various downstream tasks and improves encoder-decoder models on tasks that require explicit computation.

Future work can address the balance between textual understanding and numerical reasoning, by refining regularization strategies to maintain the language model's core strengths while enhancing its computational abilities. Tool-use in encoder-only models could also be extended to more complex tools similar to decoder-only models.

Acknowledgments

We thank Robert Lo for the helpful discussions.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038*.

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Nquad: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2925–2929.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600k: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Tora: A tool-integrated reasoning agent for mathematical problem solving.

Marek Kadlčík, Michal Štefánik, Ondřej Sotolář, and Vlastimil Martinek. 2023. Calc-x and calcformers: Empowering arithmetical chain-of-thought through interaction with symbolic systems. In *Proceedings of the The 2023 Conference on Empirical Methods in Natural Language Processing: Main track*, Singapore, Singapore. Association for Computational Linguistics.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 1152–1157.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Georgios P Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. *arXiv preprint arXiv:1805.08154*.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019. Quarel: A dataset and models for answering questions about qualitative relationships. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7063–7071.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. *arXiv preprint arXiv:1909.07940*.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854, Copenhagen, Denmark. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. Do language embeddings capture scales? *arXiv preprint arXiv:2010.05345*.

AISPACE at SemEval-2024 task 8: A Class-balanced Soft-voting System for Detecting Multi-generator Machine-generated Text

Renhua Gu, Xiangfeng Meng
Samsung R&D Institute China-Beijing
{renhua.gu, xf.meng}@samsung.com

Abstract

SemEval-2024 Task 8 provides a challenge to detect human-written and machine-generated text. There are 3 subtasks for different detection scenarios. This paper proposes a system that mainly deals with Subtask B. It aims to detect if given full text is written by human or is generated by a specific Large Language Model (LLM), which is actually a multi-class text classification task. Our team AISPACE conducted a systematic study of fine-tuning transformer-based models, including encoder-only, decoder-only and encoder-decoder models. We compared their performance on this task and identified that encoder-only models performed exceptionally well. We also applied a weighted Cross Entropy loss function to address the issue of data imbalance of different class samples. Additionally, we employed soft-voting strategy over multi-models ensemble to enhance the reliability of our predictions. Our system ranked top 1 in Subtask B, which sets a state-of-the-art benchmark for this new challenge.

1 Introduction

Large Language Models (LLMs) have emerged as a foundational element for artificial intelligence (AI) applications. Their text generation capabilities are impressive and have almost reached the human-level performance. However, their widespread use also poses risks. The use of LLM-generated text can lead to the spread of inaccurate information, academic dishonesty, and privacy breaches. Additionally, the machine-generated text may become trapped in a loop during the process of LLMs' own development, gradually replacing human-written training data and reducing the quality and diversity of subsequent models(Wu et al., 2023). To prevent the misuse of LLMs and improve the iterative refinement of AI tools, it is crucial to distinguish between machine-generated and human-written text. SemEval-2024 Task 8: Multigenerator, Multidomain and MultiLingual Black-Box Machine-Generated Text Detection(Wang et al., 2024) introduces the task of detecting machine-generated text across various generators, domains and languages. Our system focuses on Subtask B, which is a multi-class classification task. It involves detecting text from multi-generators over multi-domains in English only. Given a text, the system tells whether the text is written by a human or generated by a particular LLM. It emphasizes not only the accuracy of detecting the in-domain texts, but also the generalization to identify other out-of-domain text sources.

Current research on LLMs text detection primarily focuses on ChatGPT or a specific model in a limited domain. Gao et al., 2023 compares scientific writing between humans and ChatGPT exclusively. Wang et al., 2023b detects AI-generated news by ChatGPT. However, there are many other emerging LLMs that generate various domain texts that needed to be distinguished from those written by humans. Wang et al., 2023a presents a large-scale corpus generated by popular LLMs, including ChatGPT, Cohere, Davinci, Bloomz, and Dolly, across various domains such as Wikihow, Wikipedia, Arxiv, PeerRead, and Reddit. To address this complex scenario, we fine-tune transformer-based encoder-only, decoder-only, and encoder-decoder models.We then ensemble the model that performs best for a specific class and mitigate sample imbalance using a weighted loss function.

Our contributions can be summarized as follows:

1) We conducted a systematic research of fine-tuning language models to detect multi-class machine-generated text.

2) We developed class-balanced loss function and soft voting model ensemble to keep model robustness and generalization.

3) Our system formulated a SOTA benchmark

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1476–1481

on the task.

2 Related Work

Researchers have explored automatic detection methods for distinguishing machine-generated text from human-written text. These methods can be categorized into two distinct groups, i.e., metric-based methods and model-based methods (He et al., 2023).

2.1 Metric-based Methods

Metric-based methods utilize metrics such as log-likelihood, word rank, and predicted distribution entropy. For example, GLTR (Gehrmann et al., 2019) is developed as a visualization tool to facilitate the labeling process of whether a text is machine-generated. DetectGPT (Mitchell et al., 2023) define a new curvature-based criterion for distinguish machine-generated text under the assumption that text sampled from an LLM tends to occupy negative curvature regions of the model's log probability function. GPT-who (Venkatraman et al., 2023) is a system that computes interpretable Uniform Information Density (UID) features based on the statistical distribution of a given text. Additionally, it autonomously learns the threshold between different authors using Logistic Regression.

2.2 Model-based Methods

On the other hand, model-based methods involve training classification models using both machine-generated text and human-written text. COCO (Liu et al., 2022) incorporates coherence information into text representations through the use of a graph-based encoding method. This approach is combined with a contrastive learning framework, and an enhanced contrastive loss function is proposed to mitigate potential performance degradation resulting from simple samples.

3 System Overview

Based on the analysis of the task situation, we have carried out preliminary studies of several methods and integrated pre-trained language models fine-tuning, class-balanced weight loss function, and soft-voting model ensemble into our system.

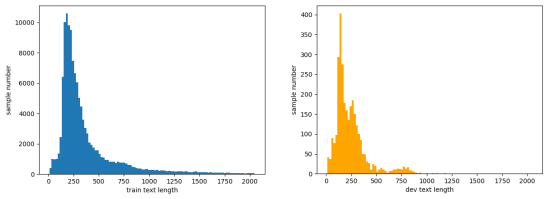
3.1 Data Process

Subtask B shares same generators, same domains and same language with subtask A. The statistical analysis reveals that subtask B lacks training data from PeerRead Source while subtask A can provide necessary data to fill the gap. To strengthen data source for training, we merged A and B train data into a unified dataset, removing any duplicated items and those present in the dev set. We then re-labeled all the texts based on task B labels. The resulting training data consists of 127,755 items. For each class, the number of sample is shown in Table 1. However, it is important to note that the training data does not include any PeerRead texts generated by BLOOMZ, unlike the dev data. We can still assess the model's generalization ability using this data.

Table 1: Sample Number of each class. C_0: human, C_1: ChatGPT, C_2: Cohere, C_3: Davinci, C_4: BLOOMZ, C_5: Dolly

C_0	C_1	C_2	C_3	C_4	C_5
63,351	13,839	13,178	13,843	9,998	13,546

Furthermore, after the merging of data, we analysed the token length of the dataset. As illustrated in Figure 1, the majority of the token length in the training text falls within the range of 0-1000, whereas the length of the development text is mostly between 0-500. So our system tested input size of 512 and 1024 tokens in Longformer model.



(a) train (b) dev

Figure 1: Token length of data

3.2 Fine-tuning Transformer-based Models

Fine-tuning pre-trained models is typically effective approach for downstream tasks(Kalyan et al., 2021). Our system utilize a series of Transformer-based models, including encoder-based, decoder-based and encoder-decoder models, to develop a multi-class classifier through fine-tuning. One purpose is to determine which architecture is better suited in such tasks. Another purpose is to construct more bases that excel in different generators, which will benefit the overall ensemble results.

3.2.1 Encoder-only

Roberta (Liu et al., 2019) is based on the architecture of BERT (Devlin et al., 2018) and incorporates several modifications and enhancements to improve its performance. One key difference with Roberta-large is its training process, which involves training on more data for a longer period of time compared to BERT. Additionally, Roberta-large uses dynamic masking during training, where the masking pattern changes from epoch to epoch, leading to better generalization. Roberta-large has demonstrated state-of-the-art performance on various NLP benchmarks and tasks, showcasing its effectiveness in understanding and processing human language. It has been widely adopted in academic research and industry applications due to its impressive results. The model was adopted as the baseline model to explore the effectiveness of our proposed methods.

Deberta (He et al., 2020) improved its performance from BERT by disentangled attention mechanism, which allows the model to focus on different aspects of the input independently, enabling better understanding of long-range dependencies and capturing complex linguistic structures more effectively. In addition, Deberta incorporates a novel masking scheme and dynamic upsampling during training, leading to improved model learning and generalization capability.

Longformer Traditional NLP models like BERT are designed to handle sequences of up to 512 tokens, limiting their applicability to longer documents such as scientific papers, legal contracts, or lengthy news articles. Longformer (Beltagy et al., 2020) includes a combination of global attention and sparse attention patterns. Global attention allows the model to capture relationships between distant tokens in the input sequence, while sparse attention reduces the computational complexity of processing long sequences. This balance enables Longformer to efficiently handle lengthy documents while maintaining strong performance. According to the analysis of token length, we fine-tuned 2 versions of this model with the input sizes as 512 and 1024 to asses the impact of input size.

3.2.2 Decoder-only

XLNet (Yang et al., 2019) builds upon the Transformer-XL (Dai et al., 2019) architecture, which includes techniques for handling long-range dependencies in sequences more effectively than standard transformer architectures. This architecture enhances XLNet's ability to capture complex relationships within text data. XLNet introduces permutation language modeling, which enables the model to capture bidirectional context without relying on the autoregressive property found in traditional models like GPT-2. This approach allows XLNet to consider all permutations of the input sequence during training, leading to a more comprehensive understanding of the contextual information.

3.2.3 Encoder-decoder

T5 (Raffel et al., 2020) belongs to the family of transformers using "text-to-text" framework, which means that it can perform a wide range of NLP tasks by converting both the input and output into text strings. This flexibility allows T5 to handle various tasks such as translation, summarization, question-answering, and more, all within a unified framework. Besides, T5 is pre-trained using a large-scale dataset and fine-tuned for specific NLP tasks, making it a highly adaptable and efficient model for a wide range of applications.

3.3 Class Balanced Weighted Loss

As shown in Table 1, each class has a different number of samples. The number of human-written samples number is even 5-6 times greater than others. To address the sample imbalance of different classes, we employed a weighted loss function during training to balance the contribution of each class sample to the loss.

For multi-class classification, the commonly used loss function is ordinary cross-entropy (CE). However when there is an imbalance-sample problem, the class-balanced weighted cross-entropy (WCE) will significantly improve the performance (Cui et al., 2019).

The weight of each class is calculated as the inverse number of samples. Denote that the number of classes is C, total number of all samples is N_{total} the number of text samples in $Class_i$ is N_i, the weight factor of each class is calculated as:

$$\{w_0, w_1, ..., w_C\} = \left\{ \frac{N_{total}}{N_0 * C}, \frac{N_{total}}{N_1 * C}, ..., \frac{N_{total}}{N_C * C} \right\}$$

3.4 Soft Voting

To enhance robustness and stability across generators and domains, we employ an ensemble approach by using the soft voting method with multiple base models.

Firstly, we obtain the confusion matrix of each base classifier. Secondly, we select the model that outperforms in a specific class. Finally, we integrate all the soft-max probability distribution matrix of all outperformed models to obtain the average probability distribution, and make the final decision based on it. The probability of the text

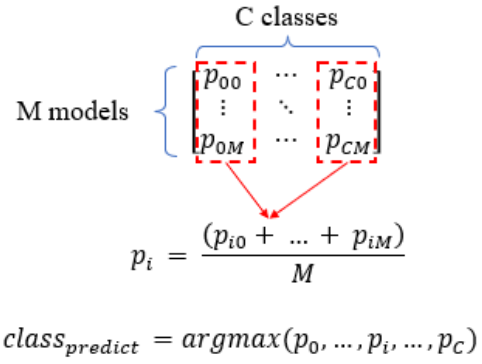


Figure 2: Soft voting over M models

belonging to $Class_i$ as predicted by $Model_j$ is denoted by p_{ij}. The final probability of $Class_i$ is calculated as the average probabilities from M models, as illustrated in Figure 2. The prediction result of the model ensemble is determined by identifying the highest probability.

4 Experimental setup

4.1 Dataset and Evaluation Metrics

For the baseline experiments on Roberta-large, we utilized the official subtaskB dataset and the merged data for separate training to determine the effectiveness of the merged data. For fine-tuning other models, we only used the merged data, which has been proven effective in the baseline experiment. The original subtaskB training data consists of 71027 items, while merging data results in a total of 127,755 items. We divided the data into train and val sets in an 8:2 ratio, and the original subtaskB dev set was kept as the dev set, which contains 3,000 items. No additional data was used for either training or evaluation. The official evaluation metric for the SubtaskB is accuracy. The experiments result in this paper is based on accuracy of dev set.

4.2 Base models

Roberta-large serves as the baseline in our system to verify the contribution of our proposed methods. We explored different datasets, loss functions, learning rates and epochs on Roberta-large to identify which are suitable for this task. Once identified, we applied them to fine-tune other models further.

There are groups of experiments on Roberta-large, settings are as follows:

- Dataset contribution: The model was fine-tuned with the original data and merged data for 3 epochs using a learning rate of 3e-5.
- Loss function contribution: The model was fine-tuned with merged data for 3 epochs with learning rate of 3e-5, applying ordinary CE and weighted CE loss in the process of training.
- Epoch contribution: The model was fine-tuned with the merged data for 3 and 5 epochs with a learning rate of 2e-5.
- Learning rate contribution: The model was fine-tuned with merged data for 5 epochs with learning rates of 1e-5, 1.5e-5, 2e-5 and 3e-5.

Deberta-large and XLNet-large applied the merged data and weighted CE as they have been shown to be effective in previous experiments. We explored various learning rates (including 1e-5, 2e-5, and 3e-5) and epochs (including 2epochs, 3 epochs, and 5 epochs), and selected the best setting as a learning rate of 1e-5 and 3 epochs as the performance comparison to other models.

Longformer is good at handling long documents, breaking the limits of 512 tokens of Bert-family models. Since we have part of long documents whose tokens numbers are greater than 512 tokens, so we tried 1024 tokens input as well as 512 tokens. Further more, to keep the pre-trained ability on semantic understanding, we fixed the top 18 layers and only fine-tuned the remained ones. Then we fine-tuned it with merged data for 5 epochs with a learning rate of 3e-5.

T5 is pre-trained on a large set of corpus and has strong adaption. We fine-tuned it with our merged data over its default parameter setting for 3 epochs with learning rates at 2e-5. Referring to the appendix of T5, a prefix (M4 sentence:) was added to each input text, then the model was trained to generate "human" or "machine".

Table 2: The performance Comparison of multiple methods on Roberta-large

Method	Epochs/LR	Accuracy
baseline	3/3e-5	0.7390
+ merged data	3/3e-5	0.9050
+ merged data + WCE	3/3e-5	0.9150
+ merged data + WCE	5/3e-5	0.9733
+ merged data + WCE	5/2e-5	**0.9800**
+ merged data + WCE	5/1.5e-5	0.9626
+ merged data + WCE	5/1e-5	0.9433

Table 3: The performance comparison of different base models

Arch	Model	Accuracy
Encoder	Roberta-large	**0.9800**
	Deberta-large	0.9730
	Longformer-512	0.9643
	Longformer-1024	0.9573
Decoder	XLNet	0.9730
Encoder -Decoder	T5	0.8617

Table 4: The performance comparison of different base model ensemble

ensemble base models	excel in						accuracy
	$Class_0$	$Class_1$	$Class_2$	$Class_3$	$Class_4$	$Class_5$	
best single model							0.9800
Roberta-large	✓	✓		✓			0.9913
Deberta-large		✓			✓		
Roberta-large	✓	✓		✓			0.9943
Deberta-large		✓			✓		
XLNet-large					✓		
Roberta-large	✓	✓		✓			**0.9946**
Deberta-large		✓			✓		
XLNet-large					✓		
Longformer	✓	✓	✓			✓	

5 Results and Analysis

To assess the efficacy of our proposed methods, we carried out multiple sets of experiments.

On the baseline Roberta-large, an ablation study was conducted. The performance comparison is shown in Table 2. The results of experiments indicate that supplementing the data source significantly improves performance. Therefore, the supervised fine-tuning is crucial in such cases. Additionally, a weighted loss function can mitigate sample imbalance issue.

Further, we fine-tuned a series of transformer-based models to select the most suitable base model. The results in Table 3 shows that the encoder or decoder can achieve top performance while the Encoder-Decoder is poor for this task. For input size, 512 tokens exceed 1024 tokens. To include longer input has no contribution to the result.

At last, we conducted a model ensemble by soft voting method to ensure robustness and generalization and reduce the effect of noise. The selected single base fine-tuned model is chosen based on its performance in the specific class. We tested various combinations, and the results are shown in Table 4. We attempted to combine various single base models, including 2, 3, and 4 types. Compared to the best single model, the ensembled model showed significant improvement, even with the least number of ensemble types. Furthermore, as the differences in ensemble models increased, the results improved even further. Additionally, if the ensemble base models perform well individually in every class, the overall result is also improved.

6 Conclusion

This paper presents a systematic study on detecting machine-generated text from multi-generators and multi-domains. We fine-tuned a series of transformer-based models and found that the encoder architecture is better suited for the task. We employed a weighted Cross Entropy loss function to address the sample imbalance. To improve robustness and generalization, various base models were ensembled by soft-voting method, and resulting in 99.46% accuracy on the dev set. In the final test, our system ranked 1st. Moving forward, we plan to explore more widely used LLMs and work towards enhancing our capabilities in few-shot learning and transfer-learning for similar tasks.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Catherine A Gao, Frederick M Howard, Nikolay S Markov, Emma C Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T Pearson. 2023. Comparing scientific abstracts generated by chatgpt to real abstracts with detectors and blinded human reviewers. *NPJ Digital Medicine*, 6(1):75.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. Ammus : A survey of transformer-based pretrained models in natural language processing.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2022. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *arXiv preprint arXiv:2212.10341*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector. *arXiv preprint arXiv:2310.06202*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023a. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Zecong Wang, Jiaxi Cheng, Chen Cui, and Chenhao Yu. 2023b. Implementing bert and fine-tuned roberta to detect ai generated news by chatgpt.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. 2023. A survey on llm-generated text detection: Necessity, methods, and future directions.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

SemEval-2024 Task 7: Numeral-Aware Language Understanding and Generation

Chung-Chi Chen
AIST, Japan
c.c.chen@acm.org

Jian-Tao Huang
Zhejiang Lab, China
jthuang@nlg.csie.ntu.edu.tw

Hen-Hsen Huang
Academia Sinica, Taiwan
hhhuang@iis.sinica.edu.tw

Hiroya Takamura
AIST, Japan
takamura.hiroya@aist.go.jp

Hsin-Hsi Chen
National Taiwan University, Taiwan
hhchen@ntu.edu.tw

Abstract

Numbers are frequently utilized in both our daily narratives and professional documents, such as clinical notes, scientific papers, financial documents, and legal court orders. The ability to understand and generate numbers is thus one of the essential aspects of evaluating large language models. In this vein, we propose a collection of datasets in SemEval-2024 Task 7 - NumEval. This collection encompasses several tasks focused on numeral-aware instances, including number prediction, natural language inference, question answering, reading comprehension, reasoning, and headline generation. This paper offers an overview of the dataset and presents the results of all subtasks in NumEval. Additionally, we contribute by summarizing participants' methods and conducting an error analysis. To the best of our knowledge, NumEval represents one of the early tasks that perform peer evaluation in SemEval's history. We will further share observations from this aspect and provide suggestions for future SemEval tasks.

1 Introduction

In the past, SemEval has predominantly focused on discussions surrounding words in text, with limited exploration of numbers in text. Recognizing the significance of understanding numbers can enhance performance in certain tasks. For instance, there is a notable difference in the sentiment degree between "expecting the stock price to increase by 30%" and "expecting the stock price to increase by 3%" in fine-grained sentiment analysis, as the former suggests a higher sentiment degree than the latter (SemEval-2017 Task 5 (Cortis et al., 2017)). Similarly, "Stealing $10" versus "Stealing $100,000" could result in differing court judgments (SemEval-2023 Task 6 (Modi et al., 2023)), and contrasting systolic blood pressure readings of 119 versus 121 offer different clinical inferences (SemEval-2023 Task 7 (Jullien et al., 2023)). These examples underscore the importance of numerical understanding in text, suggesting it as a potential research direction for enhancing the performance of downstream tasks.

Recent interest has surged in the numeracy of textual data and models within the NLP community, marking an opportune moment to evaluate current models' performance in numeral-aware language understanding and generation. To this end, we propose a collection of five published datasets encompassing three tasks: quantitative understanding, reading comprehension of numerals in text, and numeral-aware headline generation. For quantitative understanding tasks, we utilize the Quantitative 101 dataset (Chen et al., 2023). The NQuAD dataset (Chen et al., 2021) serves to explore reading comprehension with numerically rich documents, and Num-HG (Huang et al., 2024), annotated for numerical reasoning, facilitates the investigation of numeral-aware headline generation. In summary, while these are foundational NLP tasks, our focus is on discussing instances that require numeracy and the capacity to understand numbers for resolution.

In this paper, we first provide an overview of the dataset and subsequently summarize the methods and performances of participants. The comparison of models and error analysis will be included. Additionally, we employ peer evaluation to annotate and evaluate the generated outputs of participants' systems. Our analysis and observations, based on the annotations from participants, will be shared. We hope this pilot trial can offer insights and share experiences for future studies planning to conduct human evaluations among different teams.

2 Tasks and Datasets

We list the dataset for each task, the size, and the corresponding license in Table 1. Quantitative 101, which is a collection of Numeracy-600K (Chen

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1482–1491

Task	Subtask	Dataset	Size	Unit	License
Quantitative Understanding	Quantitative Prediction (QP)	Quantitative 101	1,200,000	Sentences	CC BY-NC-SA 4.0
	Quantitative Natural Language Inference (QNLI)		9,606	Sentence Pairs	MIT License
	Quantitative Question Answering (QQA)		807	Questions	ODC-By
Reading Comprehension of the Numerals in Text		NQuAD	71,998	News	CC BY-NC-SA 4.0
Numeral-Aware Headline Generation	Numerical Reasoning Headline Generation	Num-HG	27,746	News	CC BY-NC-SA 4.0

Table 1: Summary of the tasks and datasets in NumEval.

Subtask	Question	Answer
QP	FED'S DUDLEY REPEATS EXPECTS GDP GROWTH TO PICK UP IN 2014, FROM [Masked] PCT POST-RECESSION AVERAGE	1
QNLI	S1: Nifty traded above 7500, Trading Calls Today S2: Nifty above 7400	Entailment
QQA	Elliot weighs 180 pounds whereas Leon weighs 120 pounds. Who has a bigger gravity pull? Option1: Elliot Option2: Leon	Option 1

Table 2: Example for each subtask in Quantitative 101.

News Article: Major banks take the lead in self-discipline. The five major banks' newly-imposed mortgage interest rates climbed to **1.986%** in May. ... Also approaching **2%** integer alert ... Up to **2.5%** ... Also increased by **0.04** percentage points from the previous month ... Prevent the housing market bubble from fully starting.
Question Stem: Driven by self-discipline, the five major banks' new mortgage interest rates are approaching nearly _____%.
Answer Options: (A) 0.04 (B) 1.986 (C) 2 (D) 2.5
Answer: (C)

Table 3: An example question in NQuAD.

et al., 2019), EQUATE (Ravichander et al., 2019), and NumGLUE Task 3 (Mishra et al., 2022). Some examples selected from these datasets are shown in Tables 2 and 3.[1] QP subtask aims to predict the magnitude of the masked number, and it is the coarse-grained setting for examining numeracy. QNLI and QQA subtasks require models to compare numbers to answer the question. RC task in NQuAD asks models to select a proper number for the question stem based on the given news article. The average of the micro-F1 score is used to evaluate the performance in Quantitative 101, and accuracy is used to evaluate the performance in NQuAD.

To go one step further, Num-HG extends the RC task in NQuAD. It provides numerical reasoning annotations to 27,746 news, and offers two subtasks, numerical reasoning and headline generation. The major goal of this task is to generate a headline that contains key numerical information in the news article. Table 4 shows an example of the Num-HG. In the numerical reasoning subtask, models need

[1]Examples in Tables 2, 3, and 4 are from the original papers.

News: At least **30** gunmen burst into a drug rehabilitation center in a Mexican border state capital and opened fire, killing **19** men and wounding **four** people, police said. Gunmen also killed **16** people in another drug-plagued northern city. The killings in Chihuahua city and in Ciudad Madero marked one of the bloodiest weeks ever in Mexico and came just weeks after authorities discovered **55** bodies in an abandoned silver mine, presumably victims of the country's drug violence. More than **60** people have died in mass shootings at rehab clinics in a little less than **two** years. Police have said **two** of Mexico's **six** major drug cartels are exploiting the centers to recruit hit men and drug smugglers, ...
Headline (Question): Mexico Gunmen Kill ____
Answer: 35
Annotation: Add(19,16)

Table 4: An annotation example in Num-HG.

to calculate the correct number of the blank part in the news headline. In the headline generation subtask, models must generate a headline based on the given news. Because each headline in the proposed Num-HG contains one number, models are expected to generate the same number as journalists. Our rationale is that the number selected by the journalists should be the most informative for summarizing the news article. Therefore, we will evaluate whether the generated number is correct or not. Additionally, we will further evaluate the generated headline by automatical metrics, such as ROUGE and BERTScore, and manual evaluation. Specifically, participants manually evaluate the system outputs from other teams.

3 Participants and Automatic Evaluation

There are 124 teams registered for NumEval, with 20 teams submitting their system description papers. This section provides an overview of the major methods employed in each paper, with detailed explorations available in the respective papers. As participants can select specific subtasks, results are reported in a fine-grained manner to encompass partial outcomes.

Table 5 presents the Quantitative 101 results. Chen et al. (2024) utilize Flan-T5 (Chung et al., 2022) with an instructional prompt across all tasks,

Team	Method	QP		QNLI					QQA	Score
		comment	headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
YNU-HPCC	Flan-T5 + Instruction Prompt	67.20	58.82	77.73	52.40	77.06	68.40	99.94	59.25	70.10
HIJLI_JU	BERT + Character Representation	-	-	-	-	-	-	-	53.70	-
MAMET	Orca2	96.12	97.65			98.85			100.00	-
Calc-CMU	Pre-Calc (RoBERTa + Operation Classification + Calculator)	-	-	73.90	58.17	82.21	78.00	100.00	61.05	-
JU United	BERT	40.00		-	-	-	-	-	-	-
Bit_numeval	Abe-7B + Human Feedback	-	-	86.99	87.25	71.36	75.20	56.68	-	-

Table 5: Automatic Evaluation — Quantitative 101.

Team	Method	Accuracy
YNU-HPCC	Randeng-T5-77M	89.71
JN666	BERT + Pre-Finetuning with Companring Number Task	79.40
CYUT	BERT + Number Augumentation + Features	77.09

Table 6: Automatic Evaluation — NQuAD.

outperforming direct applications of pre-trained models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and LinkBERT (Yasunaga et al., 2022). Sengupta et al. (2024) emphasize the significance of number representation in character format. Kalantari et al. (2024) employ Orca2 (Mitra et al., 2023) with fine-tuning and Chain-of-Thought (CoT) prompting (Wei et al., 2022), achieving high performance across most tasks. Veerendranath et al. (2024) introduce the Pre-Calc approach, which incorporates operation classification tasks during RoBERTa training and utilizes this knowledge to decide on calculator usage for results, highlighting the value of tool utilization. Saha (2024) experiment with BERT, while Liang et al. (2024) leverage the Abe-7B model enhanced by human feedback during training, surpassing several large language models (LLMs). In summary, findings from Quantitative 101 suggest that learning calculator usage and character-format number representation can aid in quantitative tasks. Additionally, employing tailored language models like Orca2 or integrating human feedback can further enhance performance.

Table 6 presents the results on the NQuAD dataset. Chen et al. (2024) achieved the highest performance using Randeng-T5-77M (Zhang et al., 2022). Liu et al. (2024), supporting previous research (Chen et al., 2023), demonstrated that pre-finetuning with a comparing numbers task could enhance performance. Lau and Wu (2024) introduced a numeral augmentation method to improve performance. In conclusion, a well-trained language model, such as Randeng-T5-77M, can achieve superior performance in reading comprehension tasks.

Table 7 presents the outcomes of the numerical reasoning task. Due to a few teams either missing the submission deadline or reporting their results in different formats, these are included in the unofficial evaluation section. For comprehensive details on their methodologies and results, their respective papers should be consulted. LLMs demonstrated commendable performance in this task, with the methodologies of the participants detailed subsequently. Fan et al. (2024) secured the highest performance with Qwen-72B-Chat (Bai et al., 2023), employing a strategy that distinguishes the input question as either a calculation or an application problem, alongside utilizing a data augmentation technique to enhance performance. Their approach incorporated two additional datasets: GSM8K (Cobbe et al., 2021) and MetaMathQA (Yu et al., 2023). Qian et al. (2024) disclosed the results of fine-tuning GPT-3.5, whereas Chen et al. (2024) applied Flan-T5 with Chain of Thought (CoT), complemented by the use of a calculator for accuracy improvement, which yielded superior results compared to direct arithmetic computations by models. Zhao et al. (2024) fine-tuned Mistral-7B (Jiang et al., 2023), achieving performance comparable to that of fine-tuned GPT-3.5. Gonzalez et al. (2024) combined the outputs of Flan T5 and GPT-3.5, whereas He et al. (2024) implemented Llama 2-7B (Touvron et al., 2023) with CoT. Additionally, Crum and Bethard (2024) utilized Flan-T5-Lamini, and Rajpoot and Chukamphaeng (2024) fine-tuned Mistral-7B. Bahad et al. (2024) reported the performance derived from prompting GPT-3.5. In conclusion, fine-tuning LLMs and a clear understanding of the task, particularly the decision on whether to employ an external calculator, are crucial for achieving enhanced performance in numerical reasoning tasks.

Table 8 displays the outcomes of headline generation tasks. Rajpoot and Chukamphaeng (2024) enhanced Mistral-7B, yielding headlines with numerals closely matching those chosen by journalists. In the reasoning subset, this approach also secures high accuracy. Chuang and Zhunis (2024) employed BART (Lewis et al., 2020) alongside a contractive learning approach, achieving superior performance in the copying subset. Compared to these

	Team	Method	Accuracy
Official	CTYUN-AI	Qwen-72B-Chat + Task Classification + Data Augmentation	0.95
	ZXQ	Finetuned GPT-3.5	0.94
	YNU-HPCC	Flan-T5 + CoT + Calculator	0.94
	NCL_NLP	Mistral-7B + CoT + Finetune	0.94
	NumDecoders	Ensemble (Flan T5 + GPT-3.5)	0.91
	Infrrd.ai	Llama 2-7B + CoT	0.90
	hc	Flan-T5-LaMini	0.88
	NP-Problem	Finetuned Mistral-7B	0.86
	AlRah	-	0.83
	Noot Noot	GPT-3.5	0.77
	Sina Alinejad	-	0.74
	StFX-NLP	-	0.60
Unofficial	VHA	DistilRoBERTa	-
	IUST-NLPLAB	GPT-3.5	-

Table 7: Automatic Evaluation — Numerical Reasoning.

	Team	Method	Num Accuracy			ROUGE			BERTScore			MoverScore
			Overall	Copy	Reasoning	1	2	L	P	R	F1	
Official	NP-Problem	Finetuned Mistral-7B	**73.49**	76.91	**67.26**	39.82	17.58	34.34	27.80	48.56	37.82	57.02
	Challenges	BART + Contrastive Learning	72.96	**82.17**	56.18	31.22	12.24	26.86	19.53	47.56	33.13	55.36
	YNU-HPCC	Flan-T5 + Instruction Tuning + Retrieved Similar Example	69.04	73.02	61.81	**48.85**	**24.68**	**44.18**	**51.55**	**50.10**	**50.38**	**60.55**
	Infrrd.ai	Llama 2-7B + RAG	65.84	68.35	61.26	46.79	22.36	42.10	51.01	47.26	49.13	59.73
	hinoki	T5-Based Title Generator	62.35	66.28	55.18	43.07	19.72	39.00	47.22	43.44	45.34	58.71
	NCL_NLP	Mistral-7B + CoT + Finetune	62.12	65.54	55.90	43.51	19.39	38.88	46.40	45.04	45.73	58.86
	NoNameTeam	-	55.72	57.68	52.13	40.65	17.26	35.75	44.26	40.39	42.32	57.74
	Noot Noot	GPT-3.5	38.39	57.48	3.63	31.47	11.14	27.28	25.39	43.98	34.54	55.56
	ClusterCore	Few-Shot Llama	38.23	51.57	13.94	33.47	11.84	28.93	31.88	42.23	37.03	56.41
Unofficial	VHA	T5	-	-	-	-	-	-	-	-	-	-

Table 8: Automatic Evaluation — Headline Generation.

teams, several groups have utilized LLMs, obtaining improved scores in ROUGE, BERTScore, and MoverScore metrics, albeit with reduced numeral precision. Chen et al. (2024) implemented Flan-T5 with an instruction tuning strategy and enhanced it by retrieving similar cases for model referencing, leading to top results across ROUGE, BERTScore, and MoverScore evaluations. He et al. (2024) applied Llama-2-7B with retrieval augmented generation (RAG), while Crum and Bethard (2024) developed a T5-based title generator.[2] Zhao et al. (2024) fine-tuned Mistral-7B using CoT, and Bahad et al. (2024) engaged GPT-3.5 through prompting. Singh et al. (2024) examined the efficacy of Llama under a few-shot learning framework. Overall, these findings suggest that while fine-tuning can enhance numeral selection accuracy, it might decrease the similarity between the generated headlines and the actual headlines.

4 Human Evaluation

4.1 Guidelines

To enhance the evaluation of generated headlines, we implement peer evaluation for the outputs from participants' systems. Participants are required to assess the models of other teams. The evaluation comprises two metrics:

- **Numerical Accuracy**: This metric evaluates the precision of numbers within the generated headlines. It aims to verify the correctness of numerical data presented in each headline. Systems are ranked based on their average scores, adhering to the following criteria:
  - Assign 2 points for fully accurate numerical data.
  - Allocate 1 point for partially accurate numbers.
  - Give 0 points for completely inaccurate or missing numbers.
- **Optimal Headline**: This assessment involves selecting the most appropriate headline from a set of nine options. Given that nine teams have submitted their outcomes for review, we substitute the outputs from the evaluating team with journalists' headlines, serving as the ground truth. The "best headline" is identified as the one that the evaluator considers most suitable for the journalist of the corresponding news article. The system receiving the highest number of votes will be awarded one point, with points accumulated for ranking purposes. If multiple systems tie with the

[2] https://huggingface.co/czearing/article-title-generator

Team	Numerical Accuracy	Optimal
Infrrd.ai	**1.81**	22
NCL_NLP	1.73	16
Challenges	1.70	10
YNU-HPCC	1.69	15
Noot Noot	1.68	11
hinoki	1.67	16
ClusterCore	1.60	**31**
NoNameTeam	1.59	12
NP_Problem	1.57	14
Ground Truth	-	28

Table 9: Human Evaluation

Operator	Ratio
Copy	23.42%
Trans	9.91%
Paraphrase	11.71%
Round	21.62%
Subtract	7.21%
Add	11.71%
Span	4.50%
Divide	4.50%
Multiply	5.41%

Table 10: Statistics of the operators present in the error sets of the top four systems for numerical reasoning.

same number of votes for first place on a given instance, each will receive one point.

4.2 Evaluation Results

Table 9 presents the outcomes of the human evaluation process. Numerical accuracy is derived from evaluating 50 instances, with each instance receiving three annotations. The determination of the optimal headline originates from the analysis of 100 instances. According to the results, He et al. (2024) secures the highest marks in terms of numerical accuracy, despite their fourth position in automatic evaluation. Furthermore, while Rajpoot and Chukamphaeng (2024) achieves the top rank in automatic evaluation, their performance is observed to be the least favorable in human assessment among all systems evaluated. An additional noteworthy observation is that Zhao et al. (2024), utilizing the same language model as Rajpoot and Chukamphaeng (2024), attains higher scores in human evaluation.

In the context of optimal headline generation, Singh et al. (2024) receives the highest score, even though it is placed at the lower end in automatic evaluation and does not exhibit exceptional performance in numerical accuracy. He et al. (2024) is ranked second in this regard, outperforming other teams. These findings suggest that Llama (2) excels in tasks related to headline generation, considering both numerical accuracy and optimal headline aspects. Given that the ground truth was also evaluated as a candidate, its score is disclosed in Table 9, where it achieves 28 points. This score is superior to most systems and marginally lower than that of Singh et al. (2024).

5 Discussion

5.1 Error Analysis

Through the examination of participant contributions, it is observed that simple numerical questions are on the verge of being effectively addressed with the selection of an optimal language model for specific tasks. In quantitative tasks, Kalantari et al. (2024) reports achieving over 96% in micro-F1 across all subtasks through the application of Qrca2. Within the NQuAD framework, Chen et al. (2024) employs Randeng-T5-77M to secure approximately 90% accuracy, while Fan et al. (2024) attains a 95% accuracy rate utilizing Qwen-72B-Chat. For the task of headline generation, numerous teams have recorded impressive scores in human evaluations, matching or surpassing the ground truth benchmarks. These findings suggest that the era may be approaching a point where traditional tasks requiring numerical understanding and generation are nearly resolved.

However, there remain several challenges for current language models. In Table 10, we provide statistics of the operators present in the error sets of the top four systems for numerical reasoning. For instance, when presented with the masked headline "Mother of 3 Gives Huge Gift to Dying Friend" based on the news:

> "When Beth Laitkep's breast cancer spread to her brain and spine, doctors realized she had limited time left. The concern arose about the future of her six children. 'If a miracle doesn't occur and I do not survive, could you take my children as your own?' she inquired of her friend Stephanie Culley, as recounted to People magazine. Culley agreed without hesitation. Consequently, Ace (aged 2), Lily (5), Dallas (10), Jaxson (11), Selena (14), and Will (15) moved in with Culley, her husband Donnie, and their three children following Laitkep's demise in May at 39. Fortuitously, Donnie, a construction worker, had constructed their home in Alton, Virginia, with ample bedrooms

	Infrrd.ai	NCL_NLP	Challenges	YNU-HPCC	Noot Noot	hinoki	ClusterCore	NoNameTeam	np_problem	Ground Truth
Infrrd.ai	-	11	9	9	**26**	3	15	3	12	12
NCL_NLP	19	-	0	13	0	23	0	20	1	**24**
Challenges	15	7	-	**22**	7	7	9	8	4	20
YNU-HPCC	**28**	15	1	-	0	18	5	12	5	16
Noot Noot	9	12	6	5	-	2	**31**	3	27	5
hinoki	8	9	11	5	**29**	-	23	4	6	5
ClusterCore	1	3	20	2	**70**	0	-	0	3	1
NoNameTeam	10	**18**	5	14	8	6	16	-	11	12
np_problem	8	8	14	15	6	7	12	10	-	**20**
Preferred	1	1	0	1	3	0	1	0	0	2

Table 11: Human Preference.

to accommodate everyone. 'She is exceedingly humble and refrains from seeking assistance,' a friend of Stephanie's informed WSET. 'She's an angel.' (This family adopts children who are facing terminal conditions.)"

Three out of four models filled the blank with 6, while one model suggested 7. This instance illustrates the difficulty models face with numerical reasoning in complex narrative contexts.

Another intricate scenario involves a report that "A 66-year-old woman, pregnant and poised to become Britain's oldest mother, remains unrepentant about her choice, asserting her feeling akin to a 39-year-old on certain days," as detailed by the Mirror. Despite the varied daily feelings of being 39 or 56, Munro, who is 8 months pregnant, disregards the media attention, emphasizing the personal nature of her pregnancy decision. However, all models incorrectly predicted 39 instead of 66 for the headline "Brit Mum-to-Be 'Younger at Heart' Than 66, She Tells Critics".

Moreover, there are instances where models simply replicate rather than approximate numbers. For example, the correct answer for the headline "Car Auctions Off for Record-Breaking $____M" is 34.7, yet model predictions included 34.6, 34.65, and 38.0, with 34.65 being directly taken from the article text. In this case, some generated results may still be correct but just not the same as ground truth.

5.2 Human Preference

Given that most models, particularly LLMs, are adept at producing fluent headlines, the pertinent discussion revolves around the selection criteria among multiple headline candidates. This section delves into analyzing optimal headline annotations based on participant feedback. Table 11 presents statistics from different teams' annotations, highlighting the diversity in human preferences towards

Aspect	Statistics
Average Length of Best Headlines	9.47 Words
Average Length of Other Selected Headlines	9.54 Words
ROUGE 1 between Best and Other	0.4373
ROUGE 2 between Best and Other	0.1951
ROUGE L between Best and Other	0.3791

Table 12: Statistics of the best headline (Best) and other selected headlines (Other).

headline recommendations. Notably, most systems were primarily favored by a single team, with the exception of Bahad et al. (2024), which garnered the highest votes from three teams. Singh et al. (2024)'s pronounced preference for Bahad et al. (2024)'s system outputs stands out. Apart from this unique instance, determining the superior system is challenging, as preferences may vary across users. Another key observation is the ground truth achieving scores comparable to those of headlines generated by various systems, suggesting that striving for verbatim replication of the ground truth may be becoming obsolete in the context of LLMs. The emphasis may shift towards assessing the quality of generated text through more subjective and nuanced measures. Furthermore, the human evaluation results depicted in Table 11 underscore the difficulty in appraising generated headlines through manual voting, given the variance in team preferences. This inquiry constitutes the inaugural research question posited by NumEval, paving the way for subsequent investigations aimed at enhancing headline generation methodologies.

To further elucidate, we present statistics in Table 12, computed based on headlines chosen by at least one annotator. Initially, it is observed that the length of the optimal headline closely mirrors that of other selected headlines. Additionally, we compute the ROUGE scores to compare the optimal headlines against others selected. We use the following two instances to illustrate our observations.

Consider the following headlines that garnered the most votes:

- Dow Falls 64 Points, Comes Within Half a Point of 20K
- Dow Stocks Soar but Fail to Reach 20,000 Mark

Headlines receiving one vote include:

- Dow Nears 20K, But Loses Momentum
- Dow Comes Within Half a Point of 20K
- Dow Closes Below 20K
- Dow Falls Short of 20K

This analysis reveals that while all headlines convey accurate information, their level of informativeness varies. For instance, the first headline specifies a 64-point decline, a detail absent in other titles.

Another noteworthy example is the headline "NBA Season Cancellations Likely to Extend Through November 28 Due to Salary," compared with:

- NBA Season in Jeopardy as Owners Push for 50-50 Revenue Split
- NBA Season Could Be Canceled Through Nov. 28
- NBA May Cancel 2 More Weeks of Season
- NBA to Cancel 2 More Weeks of Season
- NBA Canceling 2 More Weeks of Games? 102 More Games Gone
- NBA Planned to Ax 102 More Games

In this scenario, the optimal headline succinctly conveys the cause (salary), consequence (game cancellations), and timeframe (through Nov. 28), whereas others mention only one or two of these elements. These examples, alongside our statistics, illustrate that brevity does not necessarily equate to superiority. A headline that encapsulates the most crucial information is often more valuable. Consequently, a further proposed open research question for future studies concerns the estimation of the informativeness of the generated headline.

6 Conclusion

In this paper, we explored the complexities of numerical understanding and generation in text, an area that has garnered increasing interest within the NLP community. By introducing and evaluating a set of tasks across diverse datasets, our work highlighted significant progress towards enhancing models' numerical comprehension and their application in practical scenarios, including quantitative analysis and numeral-aware headline generation. Our comprehensive evaluation, encompassing both automatic and human assessments, demonstrated the capabilities and limitations of current methodologies, emphasizing the sophisticated understanding necessary to effectively manipulate and interpret numerical information in textual formats. As we approach the mastery of simple numerical questions with the appropriate selection of language models, our research indicates a shift towards more intricate and nuanced challenges in numerical NLP. The advancements facilitated by NumEval set the stage for future investigations into the deeper integration of numeracy and language, aiming not only for models that comprehend numbers but also for those capable of reasoning, inferring, and generating text that accurately reflects the quantitative dimensions of the world.

Limitation

Although we strive to provide a comprehensive analysis, several limitations exist in NumEval and this paper. First, for the automatic evaluation, the metrics for Quantitative 101, NQuAD, and Numerical Reasoning tasks are overly simplistic, failing to verify whether models truly engage the correct reasoning steps. Second, the numerical accuracy component of human evaluation was not annotated by a consistent group of annotators, potentially subjecting the results to variability due to the subjective nature of the task. Moreover, the selection of optimal headline candidates varies across teams since we exclude headlines generated by the annotator team's system, which may further introduce inaccuracies in human evaluation. Third, although our findings suggest the tasks appear almost solved, this perception may stem from the simplistic settings of the datasets. Our error analysis reveals ongoing challenges in complex contexts, and the discussion of NumEval omits more complex reasoning steps.

Acknowledgments

This work was supported by National Science and Technology Council, Taiwan, under grants MOST 110-2221-E-002-128-MY3, NSTC 112-2634-F-002-005 -, and Ministry of Education (MOE) in Taiwan, under grants NTU-112L900901. The work of Chung-Chi Chen was supported in part by JSPS KAKENHI Grant Number 23K16956 and a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

Sankalp Bahad, Yash Bhaskar, and Parameswari Krishnamurthy. 2024. Nootnoot at semeval-2024 task 6: Hallucinations and related observable overgeneration mistakes detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 951–955, Mexico City, Mexico. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. NQuAD: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2925–2929.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics.

Kaiyuan Chen, Jin Wang, and Xuejie Zhang. 2024. Ynuhpcc at semeval-2024 task 7: Instruction fine-tuning models for numerical understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 960–968, Mexico City, Mexico. Association for Computational Linguistics.

Hao-Yun Chuang and Ali Zhunis. 2024. Challenges at semeval 2024 task 7: Contrastive learning approach on numeral-aware language generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1670–1673, Mexico City, Mexico. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. SemEval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 519–535, Vancouver, Canada. Association for Computational Linguistics.

Hinoki Crum and Steven Bethard. 2024. hinoki at semeval-2024 task 7: Numeval task 3: Numeral-aware headline generation (english). In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 34–39, Mexico City, Mexico. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuming Fan, Dongming Yang, and Xu He. 2024. Ctyunai at semeval-2024 task 7: Boosting numerical understanding with limited data through effective data alignment. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 47–52, Mexico City, Mexico. Association for Computational Linguistics.

Andres Gonzalez, Md Zobaer Hossain, and Jahedul Alam Junaed. 2024. Numdecoders at semeval-2024 task 7: Flant5 and gpt enhanced with cot for numerical reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1250–1258, Mexico City, Mexico. Association for Computational Linguistics.

JiangLong He, Saiteja Tallam, Srirama Nakshathri, Navaneeth Amarnath, Pratiba KR, and Deepak Kumar. 2024. Infrrd.ai at semeval-2024 task 7: Rag-based end-to-end training to generate headlines and numbers. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 927–938, Mexico City, Mexico. Association for Computational Linguistics.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2024. NumHG: A dataset for number-focused headline generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023. SemEval-2023 Task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Mahmood Kalantari, Mehdi Feghhi, and Taha Khany Alamooti. 2024. Mamet at semeval-2024 task 7: Supervised enhanced reasoning agent model. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1047–1052, Mexico City, Mexico. Association for Computational Linguistics.

Tsz-Yeung Lau and Shih-Hung Wu. 2024. Cyut at semeval-2024 task 7: A numerals augmentation and feature enhancement approach to numeral reading comprehension. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 566–572, Mexico City, Mexico. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Xinyue Liang, Jiawei Li, Yizhe Yang, and Yang Gao. 2024. Bit_numeval at semeval-2024 task 7: Enhance numerical sensitivity and reasoning completeness for quantitative understanding. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1842–1853, Mexico City, Mexico. Association for Computational Linguistics.

Xinyi Liu, Xintong Liu, and Hengyang Lu. 2024. Jn666 at semeval-2024 task 7: Numeval: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 484–489, Mexico City, Mexico. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Ashutosh Modi, Prathamesh Kalamkar, Saurabh Karn, Aman Tiwari, Abhinav Joshi, Sai Kiran Tanikella, Shouvik Kumar Guha, Sachin Malhan, and Vivek Raghavan. 2023. SemEval-2023 task 6: LegalEval - understanding legal texts. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2362–2374, Toronto, Canada. Association for Computational Linguistics.

Zhen Qian, Xiaofei Xu, and Xiuzhen Zhang. 2024. Zxq at semeval-2024 task 7 fine-tuning gpt-3.5-turbo for numerical reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 218–223, Mexico City, Mexico. Association for Computational Linguistics.

Pawan Rajpoot and Nut Chukamphaeng. 2024. Team np_problem at semeval-2024 task 7: Numerical reasoning in headline generation with preference optimization. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 702–706, Mexico City, Mexico. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Samiran Saha. 2024. Ju united at SemEval-2024 Task 7: Predicting numeral using fine-tuned bert models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Partha Sarathi Sengupta, Sandip Sarkar, and Dipankar Das. 2024. Hijli_ju at semeval-2024 task 7: Enhancing quantitative question answering using fine-tuned bert models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 278–283, Mexico City, Mexico. Association for Computational Linguistics.

Monika Singh, Sujit Kumar, Tanveen, and Sanasam Ranbir Singh. 2024. Clustercore at semeval-2024 task 7: Few shot prompting with large language models for numeral-aware headline generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1730–1737, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vishruth Veerendranath, Vishwa Shah, and Kshitish Ghate. 2024. Calc-cmu at semeval-2024 task 7: Precalc - learning to use the calculator improves numeracy in language models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1479–1486, Mexico City, Mexico. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Jiaxing Zhang, Ruyi Gan, Junjie Wang, Yuxiang Zhang, Lin Zhang, Ping Yang, Xinyu Gao, Ziwei Wu, Xiaoqun Dong, Junqing He, Jianheng Zhuo, Qi Yang, Yongfeng Huang, Xiayu Li, Yanghan Wu, Junyu Lu, Xinyu Zhu, Weifeng Chen, Ting Han, Kunhao Pan, Rui Wang, Hao Wang, Xiaojun Wu, Zhongshen Zeng, and Chongpei Chen. 2022. Fengshenbang 1.0: Being the foundation of chinese cognitive intelligence. *CoRR*, abs/2209.02970.

Junzhe Zhao, Yingxi Wang, Huizhi Liang, and Nicolay Rusnachenko. 2024. Ncl_nlp at semeval-2024 task 7: Cot-numhg: A cot-based sft training strategy with large language models for number-focused headline generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 261–268, Mexico City, Mexico. Association for Computational Linguistics.

UCSC NLP at SemEval-2024 Task 10: Emotion Discovery and Reasoning its Flip in Conversation (EDiReF)

Steven Au, Decker Krogh, Esha Ubale, and Neng Wan
University of California, Santa Cruz
Baskin School of Engineering, Natural Language Processing
{sttau, dkrogh, eubale, newan}@ucsc.edu

Abstract

We describe SemEval-2024 Task 10: EDiReF consisting of three sub-tasks involving emotion in conversation across Hinglish code-mixed and English datasets. Subtasks include classification of speaker emotion in multiparty conversations (Emotion Recognition in Conversation) and reasoning around shifts in speaker emotion state (Emotion Flip Reasoning). We deployed a BERT model for emotion recognition and two GRU-based models for emotion flip reasoning[1]. Our model achieved F1 scores of 0.45, 0.79, and 0.68 for subtasks 1, 2, and 3, respectively.

1 Introduction

Emotion recognition in natural language provides quantifiable insights into the traditionally qualitative realm of emotive language, bridging fields such as psychology, cognition, and linguistics. The explosion of textual data in recent years from social media platforms like Twitter and the introduction of highly cable text-processing models has provided researchers the opportunity to perform analyses on conversations that are highly complex. Despite these developments, the inherent subjectivity of emotion continues to present a challenge to the field.

Without visual information and speech audio, NLP systems must decipher rapid changes in emotional states solely through text, missing out on the nuanced non-verbal cues that often signal these shifts during spoken interactions. The absence of these cues can lead to model inaccuracies during conversational transitions such as from joy to sarcasm, or from calmness to anger.

Understanding the dynamics of emotion in the context of conversations is vital for building better conversational agents. While classifying changes in the emotion of a speaker is an important first step in this goal, it comes up short of being able to explain why the change occurred. Emotion flip reasoning is a task which has been proposed which seeks to identify the specific cause of speaker emotion flips in the context of a conversation (Kumar et al., 2022) (Kumar et al., 2024b). For example if a speaker's emotion in one utterance is joy but in their next utterance it is sad, we would like to pinpoint which utterances in the conversation caused it whether it be another speaker's or their own.

1.1 Hinglish

Hinglish, a blend of Hindi and English written in the Roman alphabet, incorporates English words into traditional Hindi contexts. This code-mixing phenomenon is becoming increasingly prevalent as English extends its influence into non-English speaking societies. Hinglish provides a challenge to models that have only been trained on English and Hindi because the model struggles to distinguish between English and Hindi words (Solorio et al., 2014). Recent work has sought to improve model performance on code-mixed dialog and new datasets have been constructed to enable these developments (Kumar et al., 2023). One goal of this research is to contribute to this work of producing models that can better understand the emotion of speakers in Hinglish.

Commonsense discernment between languages plays a pivotal role in emotion recognition within code-mixed languages, as it aids in navigating the nuanced linguistic landscapes that arise when languages intertwine (Kumar et al., 2023). Historically, individual words and phrases have been identified as significant emotional triggers, serving as fundamental elements in the computational understanding of emotions (Mohammad and Turney, 2010). This is especially pertinent in code-mixed contexts where the semantic layers are compounded by the interplay of distinct linguistic systems.

A large number research on emotion recognition

[1] https://github.com/deckerkrogh/semeval-2024-10

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1492–1497

to date has focused on extracting and interpreting common emotion-laden lexicon from Twitter corpora. While beneficial, this approach predominantly captures public, social media-expressed sentiments, which may not fully encapsulate the subtleties found in personal or private conversational contexts. This is particularly true for code-mixed interactions, where cultural contexts and language mixing patterns can greatly affect emotional expression. Datasets produced from television shows allow for insight into these more these private conversational contexts.

2 Task Description

The **EDiReF shared task at SemEval 2024** is an amalgamation of three subtasks tasks-

(i) Emotion Recognition in Conversation (ERC) in Hindi-English code-mixed conversations,

(ii) Emotion Flip Reasoning (EFR) in Hindi-English code-mixed conversations, and

(iii) EFR in English conversations.

ERC Definition: Given a dialogue, ERC aims to assign an emotion to each utterance from a predefined set of possible emotions.

EFR Definition: Given a dialogue, EFR aims to identify the trigger utterance(s) for an emotion flip in a multi-party conversation dialogue.

Speaker	Utterance	Emotion	Trigger
Sp1	Aaj to bhot awful day tha!		
	(I had an awful day today!)	Sad	0
Sp2	Oh no! Kya hua?		
	(Oh no! What happened?)	Sad	0
Sp1	Kisi ne mera sandwich kha liya!		
	(Somebody ate my sandwich!)	Sad	0
Sp2	Me abhi tumhare liye new bana deti hun!		
	(I can make you a new one right now!)	Joy	1
Sp1	Wo great hoga! Thanks!		
	(That would be great! Thanks!)	Joy	0

Table 1: Example of a dialogue from the MaSaC dataset.

We are one of 84 teams which submitted an entry to the task 10 code submission, and one of 21 which submitted papers for the task 10 workshop. (Kumar et al., 2024a).

2.1 Datasets

Two datasets were used in this task.

MELD is a dataset released in 2017 made from dialog from the TV show *Friends*. This dataset contains a list of conversations, each with multiple utterances that have been tagged with an emotion label. It has seen extensive use in research related to emotion recognition and its use in the finetuning of transformers has produced models with major improvements in tasks such as emotion recognition. For this task the task organizers produced a modified MELD dataset which has been labeled with emotion triggers (Kumar et al., 2024b). This was used for task three.

MaSaC is a Hinglish dataset produced in 2021 containing conversations with emotion-labelled utterances which were extracted from the television show *Sarabhai vs. Sarabhai*. (Bedi et al., 2021). MaSaC was used for tasks one and two. The task organizers tagged emotion triggers for the dataset used in task two.

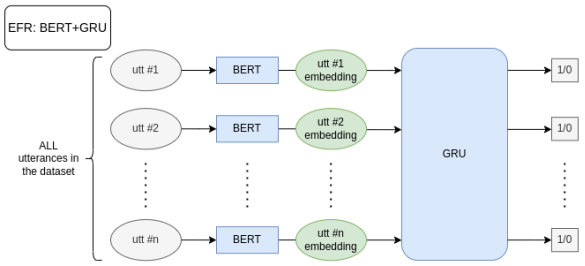


Figure 1: Illustration of the BERT+GRU architecture for the Emotion Flip Reasoning task

3 System Overview

In this study, we introduce an integrated framework that combines Bidirectional Encoder Representations from Transformers (BERT) with Gated Recurrent Unit (GRU) networks.

3.1 BERT for Emotion Recognition in Conversation

BERT was used to perform emotion recognition for the ERC task. Unlike traditional models that process text sequentially, BERT examines text bidirectionally, allowing for a comprehensive understanding of word semantics in context. Additionally, BERT's pre-training on extensive language corpora equips it with a broad understanding of language nuances, idioms, and the varied syntax used to express emotions, providing a robust starting point for fine-tuning emotion-specific datasets.

3.2 BERT+GRU for Emotion Flip Reasoning

Upon extracting contextual embeddings from BERT, we employ a GRU layer to analyze the sequence of conversational utterances. GRUs are a type of recurrent neural network optimized for

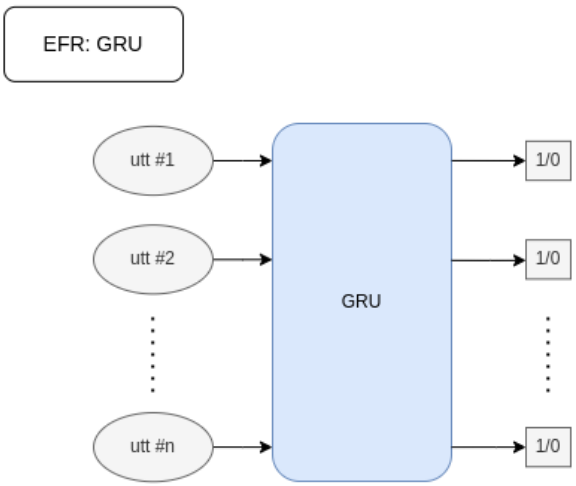


Figure 2: Simple GRU model for Emotion Flip Reasoning.

handling sequential information while mitigating issues related to long-term dependency recognition. The intent is that GRUs will be able to track the evolution of emotional states across a dialogue. Understanding the temporal sequence and the transition between emotional states is necessary in the context of emotion flip detection.

3.3 Rationale for Architectural Integration

The decision to integrate BERT with GRU stems from a strategic consideration of their respective strengths in handling different aspects of emotion analysis. BERT's contextual embeddings provide a snapshot of the emotional landscape within each utterance. While BERT excels at static context understanding, it cannot provide the sequence-to-sequence operation necessary to perform trigger classification. The purpose of the GRU is to use these static BERT embeddings to interpret the flow and dynamics of emotions through time so in order to perform trigger classification.

3.4 GRU

In addition to the BERT+GRU architecture, we also created a simpler GRU model which takes the utterances directly as input.

4 Experimental Setup

4.1 Emotion Recognition Model: BERT

We employ the pretrained BertForSequenceClassification model from the Hugging Face Transformers library. We added a fully connected linear layer with an output that matches the number of emotions. There are 7 emotions for MELD and 8 for MaSaC. The model was finetuned for 4 epochs with the AdamW optimizer set to a learning rate of 5^{-4} and an epsilon of 1^{-5}.

4.2 EFR: BERT+GRU

We constructed a deep learning model utilizing the Keras framework tailored for binary classification tasks.

BERT Embeddings: We first generate embeddings for each utterance in a conversation which will then be fed into the GRU. These embeddings were generated with the same pretrained BERT model used in the ERC task. The goal is that these embeddings can capture and provide emotion-specific information for the GRU in trigger classification.

GRU: The model consists of two bidirectional GRU layers with 32 units. We did not perform any separation between the conversations. Conversational structure is collapsed into a long sequence of utterances and passed into the GRU.

Classifier Layer: The final layer is a dense classifier with a single output unit which performs binary trigger classification for each utterance.

4.3 EFR: GRU

This model is the same as the GRU+BERT model, however instead of using BERT embeddings we use a default Keras embedding layer and pass utterances in directly.

5 Results

Table 2: F1 Scores and Task Placement

Task 1	Task 2	Task 3
0.45 (8)	0.79 (2)	0.68 (8)

5.1 Sub Task-1: ERC in Hindi-English Code-mixed Conversations

The model obtained an F1 score of 0.45 on emotion recognition in Hindi-English code-mixed conversations. The model showed a mediocre ability to capture emotional expressions.

5.2 Sub Task-2: EFR in Hindi-English Code-mixed Conversations

The GRU-only model demonstrated strong performance, achieving an F1 score of 0.76 on the validation Set and 0.79 on the test Set. This was significantly higher than the BERT+GRU model which achieved an F1 score of 0.66. These results suggest that the BERT embeddings were not able provide useful context for the GRU. It also suggests that the GRU is capable of effectively capturing the dynamic nature of emotional transitions. Despite our more novel model performing worse than the simpler one, the simple GRU achieved second place in the CodaLab competition.

5.3 Sub Task-3: EFR in English Conversations

The GRU-only model achieves F1 scores of 0.68 for the validation and 0.67 on the test set on the EFR task, outperforming the BERT+GRU model.

5.4 Further Testing

We tested additional inputs where we passed the speaker information to see if emotion recognition improved in subtasks 2 and 3 for the task. No improvements were seen in F1 but might influence the embeddings. We didn't test the EFR pipeline unless we saw improvement in ERC. We also double-checked the abnormally high emotion recognition F1 score for subtask 2 as we stripped the conversation structure and passed in duplicate utterances with shuffling. We redid the test with unique utterances and achieved .87 F1 in the test set. Surprisingly this did not affect our score for subtasks one or three. We also increased the dataset size by combining datasets 1 and 2 for Hinglish and using the whole dyadic conversations from MELD with dataset 3. The scores for ERC show no changes.

Further testing is necessary to investigate why the GRU-only model outperformed BERT+GRU. We hypothesize that it may be that the GRU simply wasn't large enough to be capable of using the the large BERT embeddings.

Another change to the model that may improve performance is to create some sort of separator embedding between the conversations. This extra information may improve performance by allowing the model to learn where triggers are placed relative to the start and end of a conversation.

6 Conclusion

The baseline approach, which employs basic embeddings of utterances and emotions, proved adequate for capturing emotion flip reasoning in large datasets, despite cultural differences and code-mixing ambiguities. The need to pass conversational information is not a substantial indicator of the prediction EFR triggers nor does passing the emotion labels into the embeddings.

In the future, we plan to explore multitask classification with BERT to determine if combined training enhances the transformer's ability to learn emotional sequence information or integrate additional conversational context to account for speaker dependencies, similar to EmoBERTa's approach. We may attempt to replicate EmoBerta's methodology to see how much emotion labels increase EFR accuracy.

Addressing the challenges of code-mixing in Hinglish and enhancing cross-cultural emotion comprehension remain critical for improving the recognition of emotional transitions.

References

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2021. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *arXiv.org*.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024a. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2024b. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, 5(3):1339–1348.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *Knowledge-based systems*, 240:108112–.

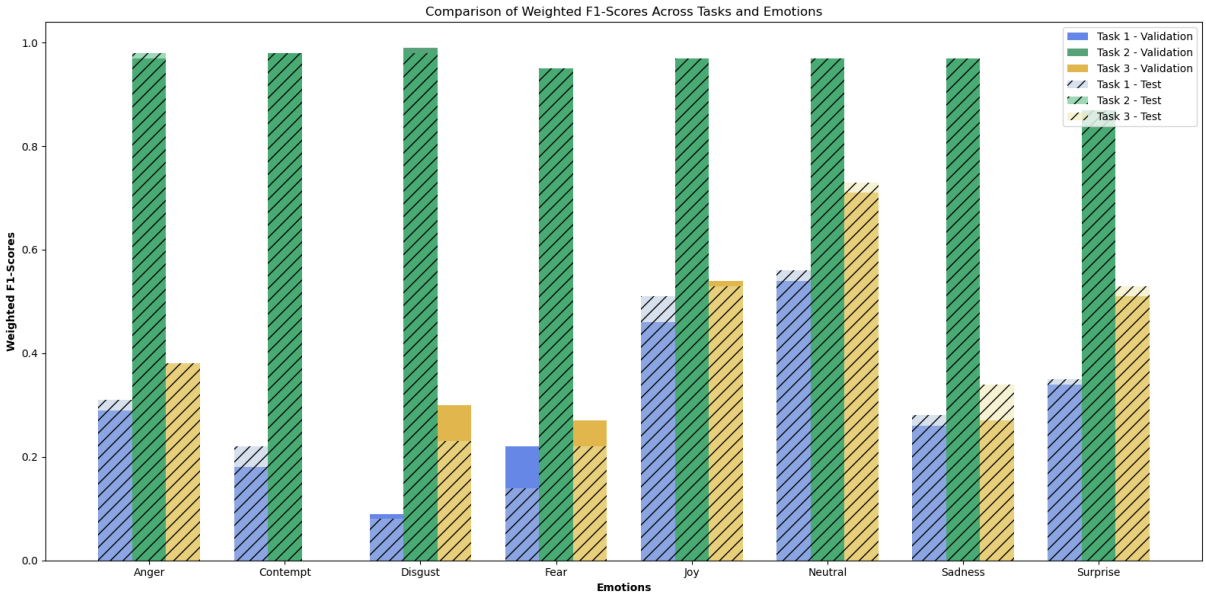


Figure 3: Comparing weighted F1 for emotions ERC across subtasks.

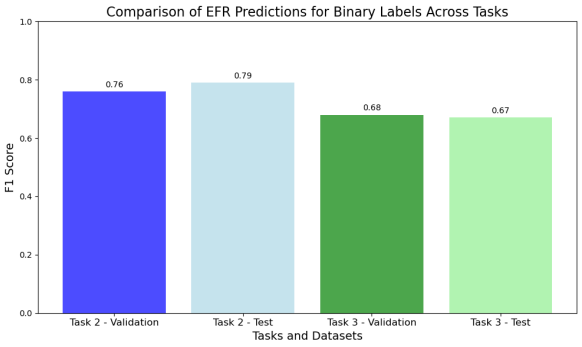


Figure 4: Blue = Subtask 2, Green = Subtask 3

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA. Association for Computational Linguistics.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

A Task Performance Metrics

Tables 4 - 13 show tables of the performance metrics across each task for the validation and test set.

Table 3: Performance Metrics for Task 1 - ERC on Validation Set

Emotion	Precision	Recall	F1-Score	Support
Anger	0.28	0.31	0.29	118
Contempt	0.22	0.15	0.18	74
Disgust	1.00	0.05	0.09	21
Fear	0.29	0.17	0.22	88
Joy	0.45	0.47	0.46	228
Neutral	0.49	0.60	0.54	633
Sadness	0.30	0.23	0.26	126
Surprise	0.31	0.39	0.34	66
Accuracy		0.48		
Macro Avg	0.43	0.30	0.31	1354
Weighted Avg	0.47	0.48	0.46	1354

Table 4: Performance Metrics for Task 1 - ERC on Test Set

Emotion	Precision	Recall	F1-Score	Support
Anger	0.31	0.31	0.31	142
Contempt	0.25	0.20	0.22	82
Disgust	0.14	0.06	0.08	17
Fear	0.18	0.11	0.14	122
Joy	0.52	0.50	0.51	349
Neutral	0.52	0.60	0.56	656
Sadness	0.33	0.25	0.28	155
Surprise	0.29	0.46	0.35	57
Accuracy		0.45		
Macro Avg	0.32	0.31	0.31	1580
Weighted Avg	0.43	0.45	0.44	1580

Table 5: Performance Metrics for Task 2 - ERC on Validation Set

Emotion	Precision	Recall	F1-Score	Support
Anger	0.98	0.97	0.97	639
Contempt	0.99	0.98	0.98	493
Disgust	0.99	0.99	0.99	87
Fear	0.98	0.92	0.95	478
Joy	0.98	0.97	0.97	1801
Neutral	0.96	0.98	0.97	3159
Sadness	0.96	0.97	0.97	487
Surprise	0.90	0.84	0.87	318
Accuracy			0.97	
Macro Avg	0.97	0.95	0.96	7462
Weighted Avg	0.97	0.97	0.97	7462

Table 6: Performance Metrics for Task 2 - ERC on Test Set

Emotion	Precision	Recall	F1-Score	Support
Anger	0.98	0.97	0.98	749
Contempt	0.99	0.98	0.98	547
Disgust	0.99	0.97	0.98	70
Fear	0.96	0.93	0.95	445
Joy	0.97	0.96	0.97	1730
Neutral	0.96	0.98	0.97	3265
Sadness	0.97	0.97	0.97	536
Surprise	0.90	0.84	0.87	348
Accuracy			0.96	
Macro Avg	0.97	0.95	0.96	7690
Weighted Avg	0.96	0.96	0.96	7690

Table 7: Performance Metrics for Task 3 - ERC on Validation Set

Emotion	Precision	Recall	F1-Score	Support
Anger	0.44	0.33	0.38	482
Disgust	0.33	0.28	0.30	64
Fear	0.26	0.28	0.27	156
Joy	0.53	0.56	0.54	597
Neutral	0.66	0.76	0.71	1360
Sadness	0.34	0.23	0.27	343
Surprise	0.51	0.51	0.51	520
Accuracy			0.55	
Macro Avg	0.44	0.42	0.43	3522
Weighted Avg	0.53	0.55	0.54	3522

Table 8: Performance Metrics for Task 3 - ERC on Test Set

Emotion	Precision	Recall	F1-Score	Support
Anger	0.46	0.32	0.38	1215
Disgust	0.36	0.17	0.23	305
Fear	0.18	0.29	0.22	177
Joy	0.50	0.56	0.53	1376
Neutral	0.71	0.76	0.73	3784
Sadness	0.36	0.32	0.34	712
Surprise	0.52	0.54	0.53	1073
Accuracy			0.57	
Macro Avg	0.44	0.42	0.42	8642
Weighted Avg	0.56	0.57	0.56	8642

Table 9: Performance Metrics for Task 2 - EFR on Validation Set

Class	Precision	Recall	F1-Score	Support
False	0.98	0.99	0.99	7028
True	0.81	0.72	0.76	434
Accuracy			0.97	
Macro Avg	0.90	0.86	0.87	7462
Weighted Avg	0.97	0.97	0.97	7462

Table 10: Performance Metrics for Task 2 - EFR on Test Set

Class	Precision	Recall	F1-Score	Support
False	0.99	0.99	0.99	7274
True	0.82	0.76	0.79	416
Accuracy			0.98	
Macro Avg	0.90	0.88	0.89	7690
Weighted Avg	0.98	0.98	0.98	7690

Table 11: Performance Metrics for Task 3 - EFR on Validation Set

Class	Precision	Recall	F1-Score	Support
False	0.94	0.96	0.95	3028
True	0.71	0.66	0.68	494
Accuracy			0.91	
Macro Avg	0.83	0.81	0.82	3522
Weighted Avg	0.91	0.91	0.91	3522

Table 12: Performance Metrics for Task 3 - EFR on Test Set

Class	Precision	Recall	F1-Score	Support
False	0.95	0.96	0.95	7473
True	0.71	0.64	0.67	1169
Accuracy			0.92	
Macro Avg	0.83	0.80	0.81	8642
Weighted Avg	0.91	0.92	0.91	8642

CLULab-UofA at SemEval-2024 Task 8: Detecting Machine-Generated Text Using Triplet-Loss-Trained Text Similarity and Text Classification

MohammadHossein Rezaei **Yeaeun Kwon** **Reza Sanayei**
Abhyuday Singh **Steven Bethard**
University of Arizona
mhrezaei,yeaeunkwon,rsanayei,abhyudaysingh,bethard@arizona.edu

Abstract

Detecting machine-generated text is a critical task in the era of large language models. In this paper, we present our systems for SemEval-2024 Task 8, which focuses on multi-class classification to discern between human-written and maching-generated texts by five state-of-the-art large language models. We propose three different systems: unsupervised text similarity, triplet-loss-trained text similarity, and text classification. We show that the triplet-loss-trained text similarity system outperforms the other systems, achieving 80% accuracy on the test set and surpassing the baseline model for this subtask. Additionally, our text classification system, which takes into account sentence paraphrases generated by the candidate models, also outperforms the unsupervised text similarity system, achieving 74% accuracy.

1 Introduction

The rapid evolution of large language models (LLMs) has significantly impacted the dynamics of information exchange, blurring the lines between human and machine-generated text. State-of-the-art LLMs are available to the public on a large scale, allowing users to generate human-like text with minimal effort. This advancement poses a dual-edged sword: while offering unprecedented capabilities in generating human-like text, it also raises critical concerns about privacy(Huang et al., 2022), ethics (Smiley et al., 2017; Kamocki and Witt, 2022), and misinformation (Pan et al., 2023; Goldstein et al., 2023; Stiff and Johansson, 2022) —especially given the LLMs' tendency to produce plausible yet factually baseless content, known as hallucinations (Dziri et al., 2022; Das et al., 2022). Distinguishing between human and machine authorship has thus emerged as a major challenge, bearing implications for content credibility and ethical standards in digital communication. As a response to the need for effective detection methods that can discern the origin of text in this new landscape, the SemEval-2024 Task 8 (Wang et al., 2024) presents an exciting challenge of AI-generated text detection over three different subtasks: Subtask A: Binary Human-Written vs. Machine-Generated Text Classification, Subtask B: Multi-Way Machine-Generated Text Classification, and Subtask C: Human-Machine Mixed Text Detection.

In this paper, we work on Subtask B, which focuses on multi-class classification to distinguish between human-written and machine-generated text by five state-of-the-art LLMs. These models are ChatGPT, `text-davinci-003`, LLaMa (Touvron et al., 2023), Cohere, Dolly-v2 (Conover et al., 2023), and BLOOMz (Muennighoff et al., 2023).

We propose three different systems to address this task: unsupervised text similarity, triplet-loss-trained text similarity, and text classification.

We show that the triplet-loss-trained text similarity system outperforms the other systems, achieving 80% accuracy on the test set and surpassing the baseline model for this subtask. Additionally, our text classification system, which takes into account sentence paraphrases generated by the candidate models, also outperforms the unsupervised text similarity system, achieving 74% accuracy. However, the unsupervised text similarity system performs poorly, achieving only 29% accuracy on the test set. We note that the latter is the only system that we submitted to the task, and the other systems are post-evaluation improvements. The main contributions of this paper are:

- An unsupervised text similarity system that computes cosine similarity to measure text similarity, which assesses the angle between vector representations of texts.
- A sentence transformer trained with triplet loss to learn the distinctions between the given texts.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1498–1504

- A RoBERTa classifier that makes decisions based on the given paragraph.
- A RoBERTa classifier which takes into account sentence paraphrases generated by the candidate models.

2 Background and Related Work

Recent research has resulted in significant advancements in Natural language generation (NLG) models (Vaswani et al., 2017) and generative pre-trained transformer (GPT) models (Devlin et al., 2019; Qiu et al., 2020) However, with potential threats posed by these models, research on identifying machine-generated text has also surged (Jawahar et al., 2020; Valiaiev, 2024). Initially, methods employing traditional machine learning models such as logistic regression were proposed (Ippolito et al., 2020). Nevertheless, the limitation of the machine learning model, which requires extensive re-training (Valiaiev, 2024), and the rise of the pre-trained transformer models, have prompted researchers to adopt the large models. Relatively smaller language models such as RoBERTa (Liu et al., 2020) have achieved state-of-the-art performance across various domains including social media, news articles, and online reviews (Uchendu et al., 2020; Adelani et al., 2020; Fagni et al., 2021). In addition to this, other approaches based on contrastive learning and similarity metrics (Boenninghoff et al., 2019) have also emerged. Such research efforts continue with the ongoing evolution and adoption of text-generative models.

3 Dataset

We work with M4 (Wang et al., 2023), a dataset for SemEval-2024 Task 8, which consists of 71,027 data samples for the training set, 3,000 data samples for the development set, and 18,000 data samples for the test set. Each sample is labeled with one of the six labels: Human, ChatGPT, Davinci, Cohere, BLOOMz, or Dolly. Figure 1 shows an example of the given dataset, which consists of id, text, model, label, and source.

Wang et al. (2023) prompted these models to write a passage given some information from the source. The sources of the texts are diverse, including Wikipedia, WikiHow (Koupaee and Wang, 2018), Reddit, arXiv, and Peer-Read (Kang et al., 2018).

4 System Overview

In this section, we provide a comprehensive overview of the three approaches we explored: unsupervised text similarity, triplet-loss-trained text similarity, and text classification.

4.1 Approach 1: Unsupervised Text Similarity

The first strategy we submitted is based on computing cosine similarity to measure text similarity, which assesses the angle between vector representations of texts. A label for multi-class classification is assigned based on the highest cosine similarity score.

4.1.1 Model Architecture

The text data is encoded using a pre-trained sentence transformer model (Reimers and Gurevych, 2019) without any additional training, followed by computing the averaged pooling embedding across all the training instances of each class. Subsequently, we calculated the cosine similarity between the text and the average-pooled embedding for each class, assigning the text to the class with the highest cosine similarity. This approach effectively categorizes the semantic similarity of texts based on topics and classifies texts with divergent writing styles (Ibrahim et al., 2023).

4.2 Approach 2: Triplet-Loss-Trained Text Similarity

Text similarity models can also be trained on the provided training data.[1] For this approach, we train a sentence transformer model with a triplet loss, which requires three inputs during training: anchor, positive, and negative samples $(x_i, {x_i}^+, {x_j}^-)$. This loss function aims to minimize the distance between the anchor and positive data $(x_i, {x_i}^+)$ while simultaneously maximizing the distance between the anchor and negative data$(x_i, {x_j}^-)$ (Ren and Xue, 2020). We conduct this training to enhance the vector representations of texts for multi-class classification.

4.2.1 Constructing Triplets

To construct the dataset with three inputs, we adopt the concept of hard positive ${x_i}^+$ and hard negative ${x_j}^-$ sampling. Hard positive involves selecting a text with the lowest similarity within the same class i, whereas hard negative involves choosing a text with the highest similarity from different

[1]This approach is a post-evaluation improvement and was not submitted to the task.

Id	Text	Model	Label	Source
557	Have you ever wanted to surprise someone with a unique and personalized cake? Look no further than an iPhone cake! With a few simple steps and some creativity, you can make a one-of-a-kind dessert that will impress anyone who sees it. Follow these steps to make your own iPhone cake: 1. Prepare 2 rectangular package cakes that can be easily form-fitted to fit with round corners. If you can't find rectangular cakes, you can simply cut and shape the cakes after baking to create the desired size and shape. . . . 10. Use several colors of fondant to create some of the apps all devices have. Work with these small pieces of colored fondants. You can use a toothpick to stick these apps into the cake, or use water and a brush to brush them onto the cake. In conclusion, making an iPhone cake is not as difficult as it may seem. This cake will be a hit with anyone, from your kids to your coworkers, and will impress them with your creativity. Just follow these simple steps and enjoy the final result!	ChatGPT	1	wikihow

Figure 1: An example of the given dataset consists of id, text, model, label, and source. Note that some part of the text from the middle is truncated with . . . for brevity.

Type	Text	Label
Anchor	How to Play Forza Motorsport This wikiHow teaches you how to play Forza ...	Human
Positive	Perfumes are a blend of different levels of scent, also called "notes". When you spray a ...	Human
Negative	Forza Motorsport is a popular racing game that provides players with the ability ...	ChatGPT

Table 1: An example of the triplet dataset which consists of anchor, positive, and negative. These pairs are chosen in a mini-batch for training. Anchor and positive data have the lowest similarity within the same class, and negative data shows the highest similarity to anchor within different classes.

classes. This concept maximizes the distinction between various classes (Robinson et al., 2021; Xu and Bethard, 2021). As the metric for similarity, we employ cosine similarity to select hard positive x_i^+ and hard negative samples x_j^- within a mini-batch. An example of the triple dataset is shown in Table 1.

4.2.2 Model Architecture

We first fine-tuned the same pre-trained sentence transformer model as in approach 1 (Reimers and Gurevych, 2019) using the triplet data and TripletMarginLoss. Then we attached a six-way classification head to the transformer using a linear layer and CrossEntropy loss. Figure 2 illustrates the overall framework, including triplet learning and classification.

4.3 Approach 3: Text Classification

We also explored a simple text classification approach where a classifier takes the given passage as the input and predicts one of the six possible labels (human or one of five LLMs) as output.[2]

We explored a variant of this text classification approach where we augment the input by asking each of the five LLMs to generate a short text. We mask a random sentence in the input paragraph and

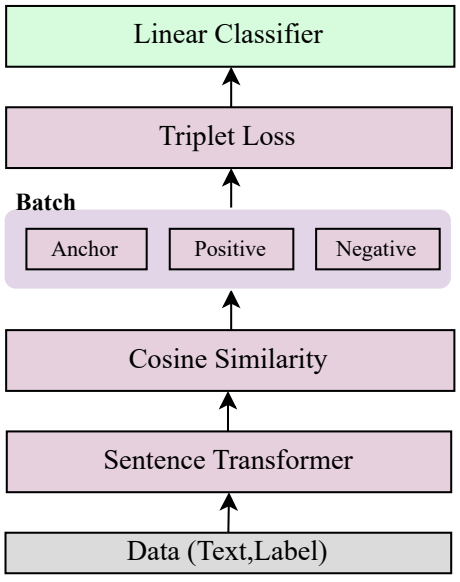


Figure 2: The overall framework of our triplet learning system proposed for Semeval-2024 Task 8.

[2]This approach is a post-evaluation improvement and was not submitted to the task.

then prompt the models to fill in the mask with a sentence that has a meaning similar to the original sentence in their own style. Due to computational limitations, we were unable to run Dolly due to its memory requirements and did not have enough resources to generate sentence paraphrases for all the over 70,000 instances in the training set. Therefore, we randomly chose 4,000 instances of each class for training, and generated paraphrases for all models other than Dolly.

4.3.1 Model Architecture

For both text classification models, we train a transformer that takes text as input and produces one of the six possible labels (human or one of five LLMs) as output. Due to the limitation of the number of input tokens for the transformer model we use (RoBERTa), we had to truncate the inputs to keep 512 tokens of the given paragraph and, for the input-augmented model, 128 tokens from each sentence paraphrase.

5 Experiments

5.1 Experimental Setup

For the unsupervised text similarity approach, we used the `paraphrase-distilroberta-base-v1` sentence transformers model from the HuggingFace library (Wolf et al., 2020). This model is based on the DistilRoBERTa architecture for clustering or semantic search. We computed the cosine similarity between the text embeddings using the PyNNDescent library[3] to facilitate an approximate nearest neighbor search in a huge dataset.

For the triplet-trained text similarity approach, we used the same sentence transformers model but trained it on the training data. We explored different hyper-parameter combinations, varying two learning rates (1-e5 and 3e-5) and two batch sizes (16 and 32) across 5 epochs and 10 epochs. Our final embedding model was trained using a learning rate of 1e-5 and a batch size of 16 for 10 epochs. For the six-way multi-class classification learning, we experimented with several classification head formulations: ReLU activation functions, dropout layers, and linear layers. The final classifier was trained with a linear layer and CrossEntropy loss. For this multi-class classification, we utilized a learning rate of 1e-5 and a batch size of 32 for 10 epochs.

[3] https://github.com/lmcinnes/pynndescent

Split	Metrics	UnSim	TripSim	TextCls	ParaCls
Test	A	0.29	0.80	0.72	0.74
Test	P	0.37	0.82	0.79	0.81
Test	R	0.29	0.80	0.72	0.74
Test	F1	0.24	0.79	0.71	0.73

Table 2: Accuracy (A), precision (P), recall (R), and F1 score of unsupervised text similarity (UnSim), triplet-trained text similarity (TripSim), text classification (TextCls), and paraphrase-augmented text classification (ParaCls).

For the text classification approach, we used the `roberta-large` model (Liu et al., 2020) from the HuggingFace library (Wolf et al., 2020). We used a learning rate of 1e-6 and 5e-7 respectively with a batch size of 8, and early stopping set to 3.

5.2 Results

Table 2 shows the performance of our different approaches – unsupervised text similarity (UnSim), triplet-trained text similarity (TripSim), text classification (TextCls), and paraphrase-augmented text classification (ParaCls) – in terms of accuracy (A), Precision (P), Recall (R), and F1 score. The submitted approach, UnSim, shows low metrics scores: 29% accuracy for the test dataset. Both the simple text classifier and the paraphrase-augmented text classifier performed better, achieving 72% and 74% accuracy on the test set, respectively. The paraphrase augmentation provided some additional information to the model, with a statistically significant improvement (McNemar's test (McNemar, 1947), $p < 0.05$) over not using sentence paraphrases. The best model was the text similarity model trained with triplet loss, which achieved 80% accuracy and 82% precision on the test dataset, surpassing the baseline model for this subtask. This improved performance underscores that the embedding obtained from triplet loss effectively learned the text distinctions by maximizing the differences between positive and negative samples.

We provide a breakdown by label for the text classification models, as shown in Table 3.

6 Conclusion

In this paper, we presented several different systems for SemEval 2024 Task 8's text classification between human and five distinct machines. Our submitted model, which relied on unsupervised embeddings coupled with cosine similarity, was poor at handling the diverse writing styles over the

Model	Label	P	R	F-1
TextCls	Human	1.00	0.44	0.61
TextCls	ChatGPT	0.52	1.00	0.68
TextCls	Cohere	0.99	0.61	0.75
TextCls	Davinvi	0.70	0.51	0.59
TextCls	BLOOMz	0.76	1.00	0.86
TextCls	Dolly	0.80	0.77	0.78
ParaCls	Human	0.99	0.39	0.56
ParaCls	ChatGPT	0.52	0.95	0.67
ParaCls	Cohere	0.97	0.66	0.78
ParaCls	Davinvi	0.77	0.64	0.70
ParaCls	BLOOMz	0.93	0.99	0.96
ParaCls	Dolly	0.67	0.80	0.73

Table 3: Detailed breakdown of results on the test set for the text classification models.

same topics that were present in the data, resulting in low classification scores. Our text classification approaches and our triplet-trained text similarity approach all outperformed the simple unsupervised model. The triplet loss learning especially improved performance over the submitted model, with its pretraining allowing it to better maximize the distinctions between texts.

For future work, we plan to adapt our systems to other classification tasks. We also plan to explore other methods for training the triplet loss model, such as using a larger model or using a different loss function. Additionally, using a larger dataset for the text classification models could improve their performance.

7 Limitations

We note that our systems are not perfect and have several limitations. For instance, we did not have enough resources to generate sentence paraphrases for all instances in the training set. We also did not have enough resources to run Dolly due to its memory requirements. Additionally, we did not explore other methods for training the triplet loss model, such as using a larger model or using a different loss function. Finally, we acknowledge that we did not try using LLMs for the classification of machine-generated text, which could potentially improve the performance of our systems.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *Advanced Information Networking and Applications*, pages 1341–1354, Cham. Springer International Publishing.

Benedikt Boenninghoff, Robert M. Nickel, Steffen Zeiler, and Dorothea Kolossa. 2019. Similarity learning for authorship verification in social media. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2457–2461.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Souvik Das, Sougata Saha, and Rohini Srihari. 2022. Diving deep into modes of fact hallucinations in dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 684–699, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Josh A. Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, and Katerina Sedova. 2023. Generative language models and automated influence operations: Emerging threats and potential mitigations.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Momen Ibrahim, Ahmed Akram, Mohammed Radwan, Rana Ayman, Mustafa Abd-El-Hameed, Nagwa El-Makky, and Marwan Torki. 2023. Enhancing authorship verification using sentence-transformers. *Working Notes of CLEF*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Pawel Kamocki and Andreas Witt. 2022. Ethical issues in language resources and language technology – tentative categorisation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 559–563, Marseille, France. European Language Resources Association.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Wang. 2023. On the risk of misinformation pollution with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1389–1403, Singapore. Association for Computational Linguistics.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Fuji Ren and Siyuan Xue. 2020. Intention detection based on siamese neural network with triplet loss. *IEEE Access*, 8:82242–82254.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*.

Charese Smiley, Frank Schilder, Vassilis Plachouras, and Jochen L. Leidner. 2017. Say the right thing right: Ethics issues in natural language generation systems. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 103–108, Valencia, Spain. Association for Computational Linguistics.

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 13(4):363–383.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Dmytro Valiaiev. 2024. Detection of machine-generated text: Literature survey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th*

International Workshop on Semantic Evaluation, SemEval 2024, Mexico, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Dongfang Xu and Steven Bethard. 2021. Triplet-trained vector space and sieve-based search improve biomedical concept normalization. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 11–22, Online. Association for Computational Linguistics.

SINAI at SemEval-2024 Task 8: Fine-tuning on Words and Perplexity as Features for Detecting Machine Written Text

Alberto J. Gutiérrez-Megías, L. Alfonso Ureña-López, Eugenio Martínez-Cámara
SINAI Research Group, Advanced Studies Center in ICT (CEATIC)
University of Jaén, Spain

Abstract

This work describes the system submitted by the SINAI team to the subtask A of Task 8 of SemEval 2024, as well as two additional systems evaluated during the training phase of the shared task. We claim that the perplexity score of a text may be used as a classification signal. Accordingly, we conduct a study on the utility of perplexity for discerning text authorship, and we perform a comparative analysis of the results obtained on the datasets of the task. The results of this study motivated us to use as classification features the word embeddings vectors of the input texts and its corresponding perplexity score. Likewise, the submitted system is a fine-tuning version of the XLM-RoBERTa-Large model. The analysis of the results of the evaluation shows large differences among the language probability distribution of the training and test sets. Nonetheless, the results show that perplexity can be used as feature for identifying machine generated text, hence our claim holds.

1 Introduction

In recent years, the use of generative models has increased considerably. The capabilities of this multifaceted tool include summarizing texts, retrieving information through searches, rephrasing texts for specific purposes and so on. However, it is important to recognize the potential threats associated with their application in certain contexts. For example, hallucinations in Natural Language Generation (NLG) models present significant problems, as they damage performance, raise safety issues for its use in the real world and hallucinations introduce privacy violation risks (Ji et al., 2023). Likewise, the very ability to generate natural language represents a threat, since it is increasingly indistinguishable from natural language. Therefore, the development of systems with the capacity to discern the authority of a given text, determining whether it is of human origin or generated by a generative model, is arising peremptory.

The language used by humans follows a probability distribution that differs so far from the distribution of the automated generated language (Rosenfeld et al., 1996). Perplexity measures the uncertainty value of a sample in a probability distribution. Accordingly, it is used in Natural Language Processing (NLP) as a metric to evaluate the effectiveness of linguistic models, for instance in text generation and machine translation tasks (Geluykens et al., 2021; Vaswani et al., 2018; Wang et al., 2019). Hence, it can be used to assess whether a text was generated by a machine, whose perplexity would be low, or written by a human, whose score would be large, since it may differ from that text automatically generated. We thus claim that perplexity can be used as a classification signal to enhance the finding of machine-generated texts (Meister and Cotterell, 2021).

In this work, we present the model submitted by the SINAI team to subtask A of task 8 of SemEval 2024 (Wang et al., 2024). Our proposal is based on the fuse of the word embeddings vectors stemmed from the fine-tuning of XLM-RoBERT-Large language model and the perplexity score of the input text. We use the Multimodal-Toolkit library (Gu and Budhkar, 2021) to fuse this two set of features.

After obtaining relevant results in the training phase and finding a clear difference between machine-generated and human-written texts, the results obtained have not been satisfactory. In part, this is due to the difference between the training and test datasets, which is analyzed later. Nevertheless, the exploration of the results obtained by merging textual content and associated perplexity raises the idea of using more novel linguistic models to calculate textual perplexity (see section 7).

The rest of the paper is organized as what follows: section 2 presents the works that support our proposal. Section 3 describes the data of the task, and section 4 is focused on the systems evaluated

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1505–1510

and the submitted one. Section 5 presents the results reached during the training phase, and Section 6 the official reached ones as well as an analysis of them. We summarize the conclusions in Section 7. We release the source code at GitHub.[1]

2 Related Work

The language generation capacity of large language models is unceasing improving (Crothers et al., 2023). Hence, machine-generated text detection is key as a fundamental countermeasure to mitigate the misuse of NLG models, accompanied by notable technical challenges and a multitude of unresolved issues.

We find a wide range of strategies to differentiate human-written text from machine-generated text (Jawahar et al., 2020) from the most simple ones based on bag-of-words models to the latest ones grounded in the fine-tuning of linguistic models.

The paper (Mitrović et al., 2023) has shown that different observable patterns make up generative models of language, either grammatically or through the meaning of sentences. For example, perplexity is usually lower in texts generated by artificial intelligence and their texts rather express feelings and use unusual words. This paper also shows a difference in performance between perplexity-based and machine learning-based classification, the latter being better than perplexity-based classification. However, it shows the capacity of perplexity score to distinguish among natural language text and machine-generated text.

3 Data and Task Description

Task 8 is focused on the identification of machine-generated text. In this work, we manage the subtasks of monolingual (English) and multilingual classification.

The dataset for the monolingual English task consists of 119,757 training instances, complemented by another 5,000 evaluation instances (Wang et al., 2023). In the multilingual task, the corpora comprise a total of 172,417 instances, with an allocation of 4,000 instances for the evaluation phase. This multilingual dataset is composed of 77.48% English text, with Bulgarian as a secondary language. The rest of the training dataset also incorporates languages such as Chinese, Indonesian, and Urdu. In addition, the evaluation dataset includes texts in Russian, German, and Arabic.

Each instance includes the *text*, along with its corresponding *source* according to five categories: *Wikihow*, *Wikipedia*, *Reddit*, *Arxiv*, *Peerread*. In the multilingual task, we can find additional sources: *Bulgarian*, *Urdu*, *Indonesian*, and *Chinese*. Also has a category that attributes the text to a specific large language model: *ChatGPT*, *Cohere*, *Bloomz*, *Davinci*, *Dolly*, or *Human* in another case. The *gold label* is 1, if the text is machine-generated and 0 otherwise. The dataset presents an even distribution, with cases annotated as human or machine being approximately equal in the training and development corpora.

4 System Description

Our proposed system to subtask A of task 8 is based on the wide success of fine-tuning methods on language models and in our claim of using perplexity as a feature to separate texts written by humans from machine-generated texts. (Min et al., 2023).

We use the XLM-RoBERTa-Large base as a language model, and we first assess its performance by fine-tuning the training data on it. Then, we evaluate the use of the perplexity as a classification signal, and the third one, which we submit to the shared-task, is based on joint use the resulting features of the fine-tuning phase and the perplexity score of each sentence.

In the next subsections, we argue the use of perplexity as feature in Section 4.1, we present all the systems studied in Section 4.2 and we describe all the implementation details in Section 4.3.

4.1 Perplexity as Feature

According to (Mitrović et al., 2023), the perplexity of human-written text tends to be higher than the one of machine-generated text. We evaluate this assertion by calculating the perplexity of the documents from the training and development sets. Table 1 shows the perplexity of texts written by humans and machines. The results for monolingual and multilingual subtasks confirm a substantial gap in perplexity in both classes, which entails that perplexity can be used as a classification signal. We use the python Language Model Perplexity library (LM-PPL)[2] to calculate the perplexity. From all the large language models available to calculate the perplexity, we use GPT2 (Radford et al., 2019).

[1] https://github.com/sinai-uja/SemEval-2024-Task-8-Identification-of-machine-written-text/tree/main

[2] https://pypi.org/project/lmppl/

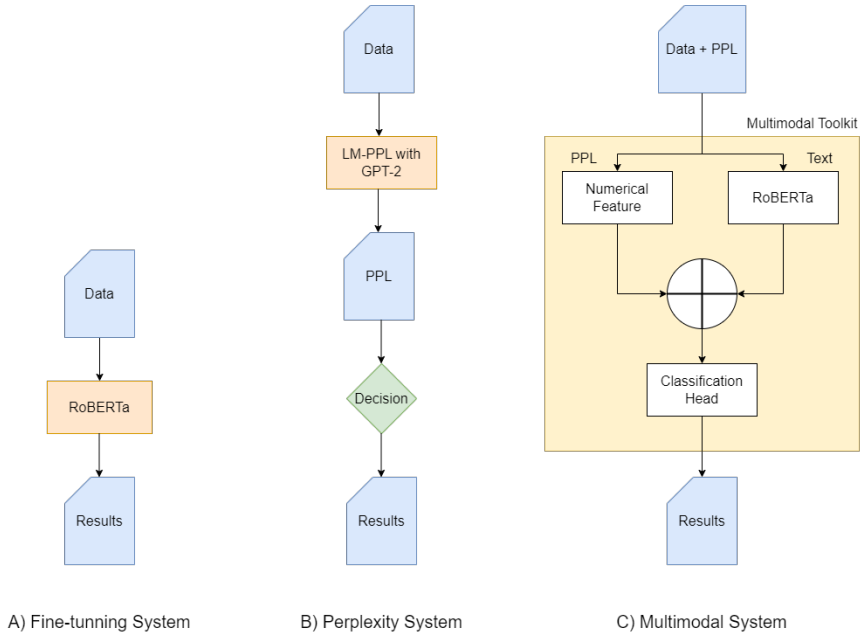


Figure 1: System diagrams used in developing phase.

Perplexity by classification	Mean
Human Monolingual	32.3613
Machine-Generated Monolingual	16.1494
Human Multilingual	35.8514
Machine-Generated Multilingual	20.9105

Table 1: Text perplexity for different classifications.

The results of Table 1 entails that the perplexity score can be use as classification signal, since it separates samples from the two classes. Accordingly, we propose a system based on the joint use of word embedding vectors and perplexity as feature.

4.2 Machine-Generated Text Detection systems

We have developed and evaluated three different systems. The first system is only based on fine tuning (system one), the second one only uses the perplexity score of the input sentences (system two) and the third one fuses the two set of features (system three). Figure 1 depicts the three systems.

System one based on fine-tuning and system two based on the use of perplexity as a classifier have been developed for the monolingual data only. Once the results of them have been obtained, the final system proposed for the task has been tested for the monolingual and multilingual subtasks.

System one - fine-tuning It is based on fine-tuning the XLM-RoBERTa-Large language model on the data of the task.

System two - perplexity For one of our systems only used the perplexity as a classifier, we established a threshold range based on the average perplexity of the dataset. The final choice of this strategy is to assign texts with perplexity at or below 20 as machine-generated, while those above this threshold are considered to be human-written.

Proposed system - fine-tuning and perplexity It is built upon two distinct features: word embedding vectors and the perplexity score of the input texts. The textual data is processed using the XLM-RoBERTa-Large transformer, and the result is sent to the combination module with the numerical features, in our case the perplexity.

The combination module uses the MultimodalToolkit library, and in particular the option[3] that separately encodes the two set of features and concatenate them before the final classification layer.

We submitted the classification results of our system based on fine-tuning and perplexity for the monolingual and multilingual subtasks.

4.3 Training and Implementation Details

We use Python to develop the proposed system and all the models evaluated during the development phase of the task. Likewise, we use the Transformers HuggingFace library (Wolf et al., 2020).

[3]Option name: *individual_mlps_on_cat_and_numerical_feats_then_concat*.

Optimized models	Epochs	Learning Rate	Weight Decay	Adam Epsilon	PP threshold
System one - fine-tuning	10	1.19e-05	6.18e-03	1.98e-07	-
System two - perplexity	-	-	-	-	20
Proposed system - Monolingual	10	6.89e-06	4.99e-02	1.13e-10	-
Proposed system - Multilingual	1	1.28e-05	8.67e-12	2.67e-07	-

Table 2: The values used for the hyperparameters of each model.

.

We optimize all the hyperparameters that drive the training of the models which involve transformers using the Optuna library (Akiba et al., 2019) following a grid search approach. This search has been performed using the English dataset, and once the optimized hyperparameters for each of the systems have been obtained, the same hyperparameters have been used for the multilingual task. For the sake of the reproducibility of the experiments, we describe the value exploration strategy of the values of the hyperparameters as what follows:

- Epochs [8, 16]: They represent the count of iterations required to traverse the entire training dataset for model training within a single cycle.
- Learning Rate [5e-6, 5e-5]: They govern the rate at which an algorithm updates or learns the parameter estimate values.
- Weight Decay [1e-12, 1e-1]: It constitutes a regularization technique that introduces a minor penalty term to the loss function.
- Adam Epsilon [1e-10, 1e-6]: It is a short positive value to forestall division by zero during the optimization process.

Table 2 shows the selected values, for the three systems that we evaluated during the development phase. We clarify that we independently optimized them, since they differ in their architecture (system one vs. proposed system) and the training objective (monolingual vs. multilingual).

5 Development results

Once we optimized the hyperparameters of each model, we assessed the performance of each system on the development data. We use accuracy as an evaluation measure, since it is the evaluation measure of the shared-task. Table 3 shows the results of this initial evaluation.

As detailed previously, the results of systems one and two are based on monolingual data, while the proposed system has been tested on both sets (see section 4.2).

System	Accuracy
System one - fine-tuning	0.8002
System two - Perplexity	0.6894
Proposed system - Monolingual	0.8698
Proposed system - Multilingual	0.6789
Task baseline monolingual	0.7400
Task baseline multilingual	0.7200

Table 3: Results in train phase with the dev. dataset.

As we indicated below, system one is only grounded in fine-tuning the XML-RoBERTa-Large model on the training data. The results are over the task baseline, which means that the incorporation of knowledge from the domain with the optimal values of the hyperparameters may reach competitive results and to overcome the task baseline.

The system that only uses perplexity as a classification signal reached poorer results than the baseline system. However, its performance is close to 70% accuracy, which means that the perplexity may be used as a feature to discriminate among human written text and machine-generated text, as we claim.

The proposed system jointly uses words and perplexity as features. In the monolingual scenario, this fusion of features reaches strong results far away from the task baseline. Nonetheless, the results for the multilingual scenario were not as strong as the monolingual one. According to the results of the monolingual data, we use as proposed system the one based on the fusion of words and perplexity as features.

6 Analysis and Discussion

The final results revealed unexpected differences compared to the results observed during the development phase. In the system that jointly uses

perplexity and text, the accuracy achieved in the monolingual task was 0.744631, which was lower than expected based on performance during development. In contrast, a poorer result was observed on the multilingual task, which achieved an accuracy of 0.801689 on the final test data set. Both results are lower than the baseline obtained by the competitors, with 0.8846 accuracy for the monolingual test and 0.8088 for the multilingual test.

Recognizing these discrepancies, our initial action consisted of an examination to determine the causes. The main advantage of our system resides in the incorporation of perplexity as classification signal along with the textual data. Consequently, our analysis primarily focused on examining the perplexity to elucidate possible factors contributing to the observed errors, as well as observing which class is more difficult to recognize, human-written or machine-generated texts.

Perplexity Performance Analysis The main limitation of our system is the use of perplexity. This metric depends on the characteristics of the reference large language model used for its calculation. Hence, we argue that there is a large disparity between the large language model used to generate the training and development sets and the documents of the test set.

We analyzed the perplexity of the documents of the test set, and we show them in Table 4. The results show a large discrepancy between the perplexity reached on the training dataset and the ones obtained on the test data. We stand out for the unexpectedly high perplexity of the machine-generated text, which is also over the human-written text. This is a sign that the large language model used to generate the documents of the test set may be more sophisticated than the one used to prepare the training dataset, or at least it was not the same large language model. This unexpected tendency to perplexity between the training and test data is behind the degradation of the performance of our proposed system on the test data, since the perplexity is a relevant feature of our porposed system.

We also explain the better performance of our proposed system in the multilingual subtask with the behavior of the perplexity on the test data. Although machine-generated text reaches again higher perplexity than human-written text, the difference is thin. Hence, the behavior of the perplexity is nearer to our claim, and the performance of our proposed system is thus stronger on multilingual data.

	Mean Perplexity	
	Human	M. Generated
Train Monolingual	32.3613	16.1494
Train Multilingual	35.8514	20.9105
Test Monolingual	35.8071	44.7824
Test Multilingual	58.4526	59.0258

Table 4: Mean perplexity in the test set for each task in comparison with the train datasets

Before generating the final results, we analyzed the prediction distribution to determine whether our system showed any tendency to predict a class in particular. In the monolingual tasks, we observed 17,978 instances of correct predictions, with only 22 false positives, with false positives being texts written by humans predicted to be machine-generated. Similarly, in the multilingual tasks, we found only 19 false negatives, the main finding was the occurrence of many false positives. Most of the predictions were obtained as machine-generated. We also highlight that the proposed system does not have any false negatives, which means that it is able to identify all the machine-generated text. However, since the disparity among the large language models to generate the training and test sets, we will keep working on reducing the false positives.

7 Conclusion

We have described the system submitted to subtask A of task 8 of SemEVAL. The system is grounded in the claim that perplexity may be a discriminant feature in identifying machine-generated texts. Hence, our submitted system is built upon the fine-tuning of a XML-RoBERTa-Large language model on a fusion of words and perplexity as features. The results reached during the development phase convinced us that our claim holds.

The official results show that the fusion of words and perplexity as features were not as good as the assessment on the development set. According to our analysis results, it may be caused by the use of a different large language model to generate the text documents. It pushes us to study the influence of the reference large language model used for the calculation of the perplexity and also to analyze the possibility of combining different perplexity calculated using a wide diverse set of large language models.

Acknowledgements

This work has been partially supported by projects CONSENSO (PID2021-122263OB-C21), MODERATES (TED2021-130145B-I00), SocialTOX (PDC2022-133146- C21) and FedDAP (PID2020-116118GA-I00) funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR".

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*.

Joppe Geluykens, Sandra Mitrović, Carlos Eduardo Ortega Vázquez, Teodoro Laino, Alain Vaucher, and Jochen De Weerdt. 2021. Neural machine translation for conditional generation of novel procedures.

Ken Gu and Akshay Budhkar. 2021. A package for learning on tabular and text data with transformers. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 69–73, Mexico City, Mexico. Association for Computational Linguistics.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *arXiv preprint arXiv:2011.01314*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Clara Meister and Ryan Cotterell. 2021. Language model evaluation beyond perplexity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5328–5339, Online. Association for Computational Linguistics.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2).

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ronald Rosenfeld et al. 1996. A maximum entropy approach to adaptive statistical language modelling. *Computer speech and language*, 10(3):187.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

USTC-BUPT at SemEval-2024 Task 8: Enhancing Machine-Generated Text Detection via Domain Adversarial Neural Networks and LLM Embeddings

Zikang Guo[1], **Kaijie Jiao**[1], **Xingyu Yao**[2], **Yuning Wan**[1], **Haoran Li**[2], **Benfeng Xu**[1]
Licheng Zhang[1], **Quan Wang**[2], **Yongdong Zhang**[1] and **Zhendong Mao**[1*]
[1]University of Science and Technology of China, Hefei, China
[2]MOE Key Laboratory of Trustworthy Distributed Computing and Service,
Beijing University of Posts and Telecommunications, Beijing, China
zdmao@ustc.edu.cn

Abstract

This paper introduces the system developed by USTC-BUPT for SemEval-2024 Task 8. The shared task comprises three subtasks across four tracks, aiming to develop automatic systems to distinguish between human-written and machine-generated text across various domains, languages and generators. Our system comprises four components: DATeD, LLAM, TLE, and AuDM, which empower us to effectively tackle all subtasks posed by the challenge. In the monolingual track, DATeD improves machine-generated text detection by incorporating a gradient reversal layer and integrating additional domain labels through Domain Adversarial Neural Networks, enhancing adaptation to diverse text domains. In the multilingual track, LLAM employs different strategies based on language characteristics. For English text, the LLM Embeddings approach utilizes embeddings from a proxy LLM followed by a two-stage CNN for classification, leveraging the broad linguistic knowledge captured during pre-training to enhance performance. For text in other languages, the LLM Sentinel approach transforms the classification task into a next-token prediction task, which facilitates easier adaptation to texts in various languages, especially low-resource languages. TLE utilizes the LLM Embeddings method with a minor modification in the classification strategy for subtask B. AuDM employs data augmentation and fine-tunes the DeBERTa model specifically for subtask C. Our system wins the multilingual track and ranks second in the monolingual track. Additionally, it achieves third place in both subtask B and C.

1 Introduction

The burgeoning capabilities of large language models (LLMs), exemplified by ChatGPT (OpenAI, 2022), GPT-4 (OpenAI, 2023) and Llama (Touvron et al., 2023a), have made machine-generated text more fluent and human-like, which has led to an increasing concern about the abuse of LLMs such as misinformation spread (Bian et al., 2023; Hanley and Durumeric, 2024; Pan et al., 2023) and disruption in education system (Perkins et al., 2023; Vasilatos et al., 2023). So far humans perform only slightly better than chance when distinguishing between text generated by LLMs and human (Mitchell et al., 2023), so it calls for an automatic system to identify machine-generated text.

To this end, MBZUAI NLP department holds SemEval-2024 Task8, which consists of three subtasks. Subtask A focuses on determining whether a full text is human-written or machine-generated. The biggest challenge lies in the domain difference between the training set and test set while the multilingual track requires strong adaptation to text across various languages. It demands the model to have a strong capability of generalization in out-of-domain scenarios. Subtask B aims at doing multi-way machine-generated text detection and brings a new challenge of identifying the text source without knowing the domains in the test set. Subtask C proposes human-machine mixed text detection, which gives a text where the first part is human-written and the second part is machine-generated. The goal is to robustly determine the boundary where the change occurs using a fairly small training set. A detailed description can be found in the task description paper (Wang et al., 2024).

Currently, training-based machine-generated text detection strategies such as fine-tuning RoBERTa model (Solaiman et al., 2019) underperform in the out-of-domain scenario. Prominent zero-shot methods (Mitchell et al., 2023; Yang et al., 2023) can only discriminate whether a text is produced by a specific LLM or by a human. Sniffer (Li et al., 2023) and SeqXGPT (Wang et al., 2023a) appear to handle the problem of subtask B and C, but the claimed result is based on the assump-

*Corresponding author: Zhendong Mao.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1511–1522

tion that we know the origins of the text. Therefore, we propose our system, which consists of **D**omain-**A**daptive **Te**xt **D**etection (**DATeD**), **L**LM-Powered **L**anguage-**A**ware **M**odel (**LLAM**), **T**hree-stage **L**LM **E**mbeddings (**TLE**) and **Au**gmented **D**eBERTa **M**odel (**AuDM**). It performs outstandingly in out-of-domain scenarios, especially in subtask A.

In DATeD, we enhance performance by innovatively incorporating Domain Adversarial Neural Networks (DANN) (Ganin et al., 2016) into the task of machine-generated text detection. DANN consists of a feature extraction layer and a category predictor, forming the backbone network to predict classification labels. Furthermore, the domain classifier is connected to the backbone network through a gradient reversal layer for classifying domain labels. This enables the model to learn transferable features between the training and development set, effectively overcoming challenges in out-of-domain scenarios. We achieve **second** place out of **126** participants.

In LLAM, we handle text from various languages in a distinct manner. For English text, we employ LLM Embeddings, leveraging the powerful representation capabilities of LLM by directly extracting embeddings from the last layer of a proxy LLM, and we classify the text using a two-stage CNN. For text in other languages, we utilize LLM Sentinel, which reframes the classification task as a next-token prediction task. We win the **first** place in the track among **59** participants.

In TLE, we utilize the LLM Embeddings method mentioned above, with only a minor difference in the classification strategy. We rank **third** out of **70** participants.

In AuDM, we fine-tune a DeBERTa-base (He et al., 2021) model with a linear layer for token classification. This is an easy but effective system and ranks **third** out of **30** submissions.

In short, our contributions are as follows:

(1) We come up with a comprehensive system for machine-generated text detection in varies scenarios (Section §3), which significantly improves the performance compared to the baseline in all subtasks (Section §5).

(2) We utilize DANN in machine-generated text detection in DATeD (Section §3.1), and employ two adaptive strategies leveraging LLM capabilities in LLAM (Section §3.2).

(3) Extensive experimental analysis demonstrates the effectiveness of DATeD and LLAM (Section §5).

2 Related Work

2.1 Detecting LLM-Generated Text

The detection of machine-generated text is often expressed as a classification task. One way to solve this problem is to use supervised learning to train classification models on datasets that contain both machine-generated and human-written text. For example, GPTZero (Tian, 2023) collects human-written text from a variety of domains, including student-written articles, news articles, and question-and-answer datasets across multiple disciplines. G3Detector (Zhan et al., 2023) claims to be a general-purpose gpt generated text detector implemented by fine-tuning RoBERTa-large (Liu et al., 2019), however, the effect of text detection generated by multiple generators will be poor. T5-sentinel (Chen et al., 2023) trains RoBERTa and T5 (Raffel et al., 2023) classifiers on the OpenGPTText dataset they built, and then uses the T5 model's ability to predict the conditional probability of the next word to classify multiple text sources. SeqXGPT (Wang et al., 2023a) introduces the sentence-level detection challenge by synthesizing a dataset containing documents that have been polished with a Large Language Model. SeqXGPT uses sequence annotation methods to train its model and selects the most frequent class as sentence class, which provides a scheme for subtask C. However, a model explicitly trained to detect machine-generated text may overfit the training distribution of its domain (Bakhtin et al., 2019), resulting in poor generalization.

In addition, (Solaiman et al., 2019) notes the surprising power of a simple zero-shot method for machine-generated text detection, which thresholds candidate paragraphs based on their average log-probability under a generative model, a powerful baseline for many zero-shot learning machine-generated text detection tasks. DetectGPT (Mitchell et al., 2023) demonstrates that text sampled from LLMs tends to occupy regions of negative curvature of the model's log-probability function. Building upon this observation, a new curvature-based criterion is defined to determine whether a paragraph is generated by a given LLM. Its outstanding performance can only be guaranteed by a large disturbance function and a large number of perturbations, so more computational resources are required.

2.2 Domain Adversarial Neural Networks

Machine learning models typically assume that the training and test sets come from the same data distribution. However, labeled data is scarce, and it is the norm for unlabeled data, which may not align with the distribution of labeled data (HassanPour Zonoozi and Seydi, 2023), to constitute the majority of the data. In the task of machine-generated text detection, there are cases where the sources and generators differ between the source domain and the target domain. For instance, in the monolingual track of subtask A, the test set may include sources and models, such as BLOOMZ (Muennighoff et al., 2022), that are not present in the training set. Across data generated by different models, significant differences may exist in content style, text length, word frequency, and other distributions. This discrepancy results in models trained on labeled training data failing to generalize well to detect text data generated by other models. The core issue addressed by domain adversarial neural networks (Ganin et al., 2016) is mitigating the impact of inconsistent data distributions between the training and test sets on the performance of machine learning models. This enables models to learn sufficiently robust text representations and reduce differences in data distributions at the representation level. Introducing domain adversarial neural networks into monolingual generated text detection enhances the model's ability to generalize and transfer across different machine-generated text models (Chen et al., 2020).

3 System Overview

Our system consists of four components: DATeD, LLAM, TLE and AuDM. Each of these components addresses one of the four tracks of the task respectively.

In subtask A, we aim to do machine-generated text detection on mono/multilingual data. In the monolingual track, the domains in test set differ a lot from ones in training set. We innovatively apply DANN to the detection in DATeD. This is achieved by adding a gradient reversal layer on top of the base model. Additionally, besides category labels, we incorporate extra domain labels into the dataset (training set: 0, development set: 1), enabling the model to learn transferable features between the training and development set (Section §3.1).

In the multilingual track, another challenge is that the model is supposed to have robust generalization capabilities to adapt to the distinct characteristics of different languages, especially low-resource languages. We propose LLAM, which employs different methods for different languages. For text identified as English, we feed it into a proxy LLM to extract embeddings from the last layer and subsequently pass it through a two-stage CNN for classification. In the case of non-English text, we redefine the classification task as a next-token prediction task (Section §3.2).

In multi-way text detection, texts originate from human, ChatGPT, Cohere, Davinci, BLOOMZ and Dolly (Conover et al., 2023). The challenge lies in distinguishing texts generated by various LLMs. Therefore we conduct a three-stage classification based on the LLM Embeddings method (TLE) mentioned in the multilingual track to better fit the scenario of multi-classification (Section §3.3).

In human-machine mixed text detection, the target is to do fine-grained detection. Inspired by the wide use of BERT (Devlin et al., 2019) in sequence labeling task, we fine-tune a DeBERTa model with data augmentation (AuDM) to classify each token in a text (Section §3.4).

3.1 Domain-Adaptive Text Detection

The overall process of Domain-Adaptive Text Detection is illustrated in Figure 1. The model comprises three components: a feature extraction layer (such as RoBERTa) acquires text representation, a category predictor determines whether the given text is machine-generated, and a domain classifier is employed to mitigate differences in data distribution between the training and development sets, thereby enhancing the generalization capability of machine-generated text detection.

The feature extractor and label predictor constitute a feedforward neural network serving as the backbone of machine-generated text detection, utilized to classify data from the source domain. The label predictor employs MLP for classification, aiming to predict labels accurately. Following the feature extractor, we append an additional branch called the domain classifier. The domain classifier is interconnected through a gradient reversal layer to classify data in the feature space, determining whether it originates from the source domain or the target domain.

Forward propagation During forward propagation, given an input text of $\boldsymbol{n}$ tokens $\boldsymbol{x} = \{x_1, \cdots, x_n\}$, we initially input the text $\boldsymbol{x}$ into the model. We opt for RoBERTa as the

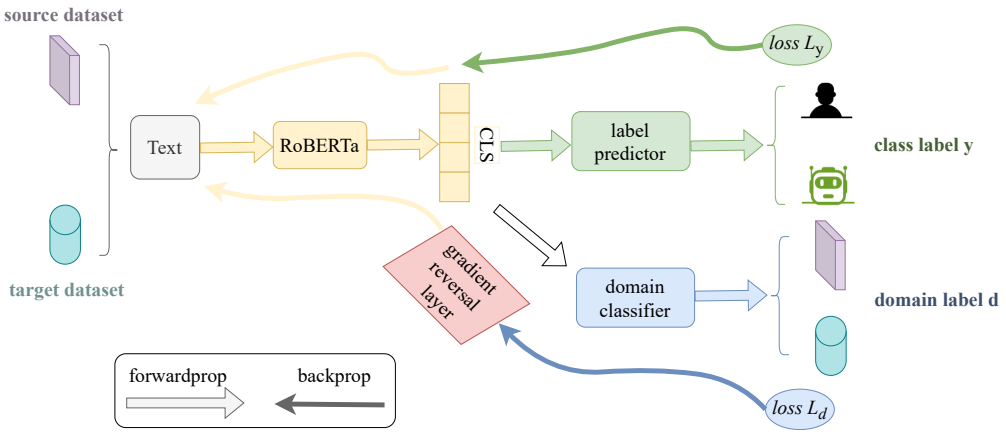


Figure 1: Overall architecture of DATeD

feature extractor $G_f(\cdot)$, utilizing the output vector corresponding to the CLS token as the semantic representation of the input text, denoted as $G_f(\boldsymbol{x})$. Subsequently, $G_f(\boldsymbol{x})$ is simultaneously fed into the label predictor $G_y(\cdot)$ and the domain classifier $G_d(\cdot)$, yielding $G_y(G_f(\boldsymbol{x}))$ and $G_d(G_f(\boldsymbol{x}))$, representing the category label $\boldsymbol{y}$ and the domain label $\boldsymbol{d}$, respectively (as shown in the green and blue sections depicted in Figure 1).

Back propagation During back propagation, the cross-entropy loss function is computed by comparing the category labels $\boldsymbol{y}$ predicted by the category predictor with the actual labels in the source domain, resulting in category loss (Ganin and Lempitsky, 2015). Additionally, we calculate the cross-entropy loss function by comparing the domain labels $\boldsymbol{d}$ classified by the domain classifier with all data from both the source and target domains, obtaining domain loss. It is worth mentioning that the gradient reversal layer behaves like a feedforward neural network during forward propagation. However, during backward propagation, the gradients are reversed (we achieved by multiplying by a negative identity matrix). Finally, the model updates its label predictor and domain classifier by summing up their respective losses (as shown in Formula 1). This setup enables the label predictor to distinguish categories in the source domain data (Ganin et al., 2016), while rendering the domain classifier unable to discern the origin domain of the data.

$$\mathcal{L}_{all} = \mathcal{L}_y + \lambda \mathcal{L}_d \tag{1}$$

$\mathcal{L}_y$ represents the label predictor loss in the source domain, $\mathcal{L}_d$ represents the domain classifier loss, λ is a hyperparameter.

3.2 LLM-Powered Language-Aware Model

In the multilingual machine-generated text detection task, we propose our model LLAM, which employs the language identification tool langdetect [1] to determine the language of the input text first. Then, we utilize LLM Embeddings and LLM Sentinel to detect English and non-English text respectively. The overall architecture of LLAM is depicted in Figure 2.

LLM Embeddings For English text, we use Llama-2-70B (Touvron et al., 2023b) as the proxy LLM to obtain embeddings of the input text. Given an input text of $\boldsymbol{n}$ tokens $\boldsymbol{x} = \{x_1, \cdots, x_n\}$, we initially input the text $\boldsymbol{x}$ into the proxy LLM to get the token embeddings from the last layer. Subsequently, we calculate the average of token embeddings $\boldsymbol{h}$ to serve as the text representation. This representation is then inputted into a two-stage CNN for classification. In the first stage, the CNN extracts relevant features from the input representation $\boldsymbol{h}$. This process is accomplished through the utilization of three convolutional and pooling layers. In the second stage, the extracted feature is fed into three fully connected linear layers to output class probabilities $\boldsymbol{p}$. The model is trained by minimizing the cross-entropy loss.

LLM Sentinel (Chen et al., 2023) have proposed utilizing the base LLM's inherent next-token prediction ability for detection, advancing the field by choosing the T5 model as the base LLM. Drawing inspiration from this approach, for non-English text, we choose the mT5-large model as our proxy LLM. LLM Sentinel relies on the LLM's capability to predict the conditional probability of the next token. Given an input text of $\boldsymbol{n}$ tokens

[1] https://pypi.org/project/langdetect/

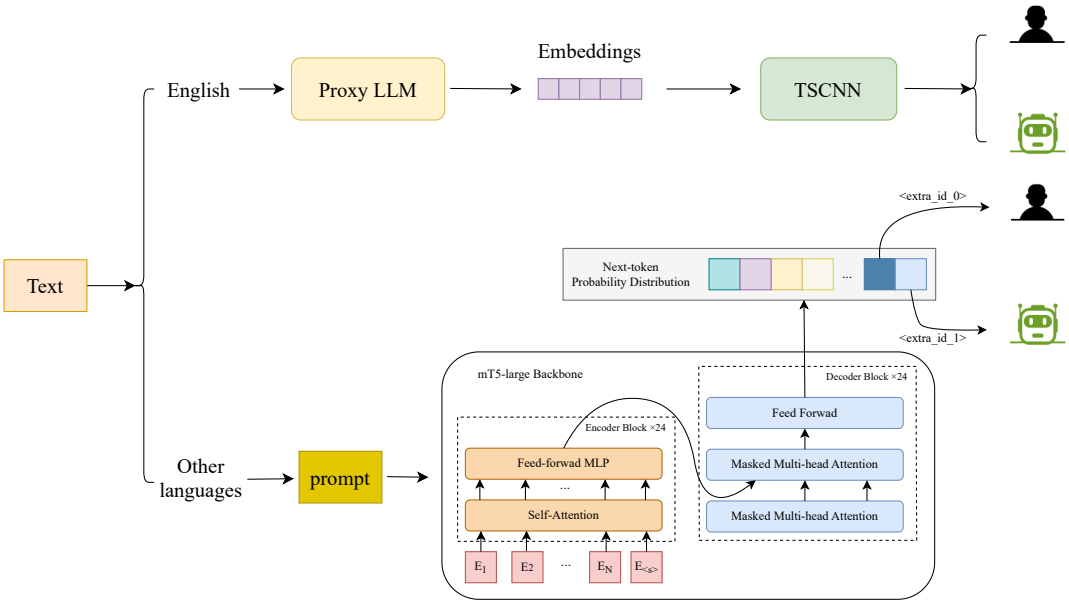


Figure 2: Overall architecture of LLAM

$\boldsymbol{x} = \{x_1, \cdots, x_n\}$, let Y denote the set of labels in this particular task, which contains "human" and "machine". We can establish a bijection $f : Y \rightarrow \mathcal{Y}$, where $\mathcal{Y}$ serves as a stand-in for the labels. Consequently, we reframe the binary classification task $\boldsymbol{x} \rightarrow Y$ as a next-token prediction task $\boldsymbol{x} \rightarrow \mathcal{Y}$. To accomplish this, we employ reserved tokens, $\mathcal{Y}$, which are not present in the text dataset. Specifically, We use $< extra\_id\_0 >$ for human and $< extra\_id\_1 >$ for machine. Therefore, the binary classification task can be effectively tackled using the LLM:

$$\hat{y} = f^{-1}\left(\arg\max_{y \in \mathcal{Y}} \mathbb{P}(y|\boldsymbol{x})\right) \tag{2}$$

Besides, in order to adapt to the characteristics of natural language, our team add a prompt before the detected text to fine-tune the LLM to perform it. The prompt used by our team is "Discern whether the following text is authored by a human or a machine. If human-written, respond with $< extra\_id\_0 >$; if machine-generated, respond with $< extra\_id\_1 >$: 'text'". Regarding the model's output, the decoding space consists of token probabilities for the entire dictionary. We simply need to extract the probability distribution over the set $\mathcal{Y}$, corresponding to the token probabilities for $< extra\_id\_0 >$ (human) and $< extra\_id\_1 >$ (machine). We then compare their magnitudes and select the larger of the two as the prediction result.

3.3 Three-stage LLM Embeddings

Building upon the LLM Embeddings approach outlined in Section 3.2, TLE additionally employs a three-stage classification process to address subtask B. Firstly, we distinguish between human-generated and machine-generated text. Subsequently, we categorize ChatGPT and Cohere as a single class for a four-class classification, differentiating them from Davinci, BLOOMZ, and Dolly. Given the challenges we encountered in distinguishing between Cohere and ChatGPT in our initial experiments, we proceed with a binary classification specifically focusing on ChatGPT and Cohere.

3.4 Augmented DeBERTa Model

In human-machine mixed text detection, we set up a model with a DeBERTa-base layer and a linear layer. The human token is classified as 0 and the machine token is classified as 1. The boundary is where the change of 0 to 1 occurs. We also perform data augmentation by employing Llama-2-7B to further generate training data.

4 Experimental setup

4.1 Datasets and Evaluation Metrics

Datasets The dataset for this task is an extension of the M4 (Wang et al., 2023b) dataset. Unlike the M4 dataset, this task samples human data to ensure data balance. New domains, generators, and languages appear in the test set to evaluate the generalization

ability of the algorithm. See Table 1 and Table 4 for the division of the dataset. See Appendix A for more details.

	Train	Dev	Test
Human	63351	2500	16272
Machine	56406	2500	18000
Total	119757	5000	34272

Table 1: Dataset division of monolingual track for subtask A.

	Train	Dev	Test
Human	83846	2000	20238
Machine	88571	2000	22140
Total	172417	4000	42378

Table 2: Dataset division of multilingual track for subtask A.

	Train	Dev	Test
Human	11997	500	3000
ChatGPT	11995	500	3000
Cohere	11336	500	3000
Davinci	11999	500	3000
BLOOMZ	11998	500	3000
Dolly	11702	500	3000
Total	71027	3000	18000

Table 3: Dataset division of subtask B.

Train	Dev	Test
3649	505	11123

Table 4: Dataset division of subtask C.

Evaluation Metrics The evaluation metrics for subtask A and B are accuracy. Accuracy is the ratio of the number of samples that the model predicts correctly to the total number of samples. Subtask C is evaluated using the MAE metric, which calculates the absolute difference between the predicted and actual boundary positions for each sample and takes the average value. The performance is better when MAE is smaller.

4.2 Training

Domain-Adaptive Text Detection In the monolingual track of subtask A, we initially define the training set and development set as the source domain and target domain respectively. Apart from the class labels provided by the dataset, we augment both the source and target domain datasets with additional domain labels. Specifically, the domain labels for the source domain samples are categorized into one class (e.g., labeled as 0), while the domain labels for the target domain samples are categorized into another class (e.g., labeled as 1). Since the overall loss computation includes both the label loss from the source domain and the domain loss from both the source and target domains, an equal number of samples from both the source and target domains is necessary to calculate the total loss. Therefore, it is imperative to balance the proportion of samples between the source and target domains. Please refer to Appendix B.1 for detailed settings.

LLM-Powered Language-Aware Model LLAM is composed of two parts: the LLM Embeddings model and the LLM Sentinel model. During the training process of our LLM Embeddings model, we split the training set into new training and development sets in a 9:1 ratio. The highest accuracy attained on the new development set during training is used to select the best checkpoint. Our final LLM Sentinel model is trained with the complete training set of this subtask and undergoes validation on the entire development set of the same subtask after each epoch. The final model chosen is the one demonstrating the optimal performance on the development set. For more details about the experiment, please refer to Appendix B.2.

Three-stage LLM Embeddings Our final model is trained in three stages, utilizing data from the corresponding categories in the subtask B dataset for each stage. See more details in Appendix B.3.

Augmented DeBERTa Model For our final submission, we do augmentation by adding the development set to the training set and using Llama-2-7B to continue generating based on training data. The checkpoint with the lowest MAE on the development set is chosen for submission. More details are in Appendix B.4.

5 Results

In this section, we report our results on all three subtasks and discuss our findings of the current

System	Accuracy
Baseline	88.46
Genaios	**96.88**
mail6djj	95.76
L3i++	85.83
QUST	84.16
USTC-BUPT (ours)	96.10

Table 5: Performance on subtask A: monolingual track.

System	Accuracy
Baseline	80.89
FI Group	95.84
KInIT	95.00
priyansk	93.77
L3i++	92.87
USTC-BUPT (ours)	**95.99**
w/o LLM Embeddings	92.03
w/o LLM Sentinel	82.05

Table 6: Performance on subtask A: multilingual track.

work. We provide the final submission results, as well as the results from several top-ranked systems.

5.1 Subtask A: Monolingual Track

In this part, we present a portion of the official results from the monolingual track and analysis of the selection process for target domain data.

5.1.1 Main Results

There are 126 teams that participate in the monolingual track. Due to the limited space, we only compare our system with the systems from teams Genaios, mail6djj, QUST and L3i++. The official results are shown in table 5. Our system achieves an accuracy of 96.10% and secures second place in the official ranking, surpassing the baseline of 88.46% by 7.64%. This indicates that adding domain adversarial neural networks solves the impact of inconsistent data distribution between the training and test set, enabling the model to learn transferable features between the two sets, thus significantly improving model performance. Upon reviewing the methods published by other participants, we find that our result (96.10%) is not far from Genaios (96.88%). Notably, while Genaios utilizes the larger Llama-2-13B model, we achieve similar performance using the smaller RoBERTa-base model.

5.1.2 Target Domain Data Selection

We conduct a series of experiments regarding qualitative and quantitative data selection. **Qualitative analysis** aims to verify whether the target domain utilizes the development set or the test set. This is because the generative models utilized in the training set and the test set may overlap. Using the test set directly as the target domain could result in texts generated by the same generative model (belonging to the same domain) being assigned different domain labels. According to our submitted results, training with the development set as the target domain results in the best performance with an accuracy of 96.10%. However, training with the test set as the target domain results in an accuracy of only 88.70%. Observing the distribution of generative models in the test set further validates these findings. The test set comprises existing models such as ChatGPT, Cohere, and also features the emergence of a new generation model, GPT-4. Thus, utilizing the test set as the target domain could lead to misdefined domain labels.

Quantitative analysis aims to explore how many repetitions of the target domain could yield better results. This is because updating the loss necessitates domain label loss from both the source domain and target domain, as discussed in Section §4.2 about DATeD, which requires an equal number of samples from each. Hence, the number of target domain samples to be duplicated needs exploration. According to our submitted results, repeating the target domain 15 times yields nearly the same number of samples as the source domain, resulting in the best performance with an accuracy of 96.10%. When repeated three times, the detection accuracy in the test set decreased to 91.75%.

5.2 Subtask A: Multilingual Track

In this part, we offer partial results from the leaderboard in the multilingual track and perform an ablation study.

5.2.1 Main Results

There are 59 teams that participate in the multilingual track. Due to the limited space, we only compare our system with the systems from teams FI Group, KInIT, priyansk, L3i++. The official results are shown in Table 6. Our system achieves the best result in the official ranking with 95.99% accuracy, surpassing the baseline by 15.10%. This showcases the powerful representational capacity of LLMs and encourages us to explore further strategies to leverage it.

System	Accuracy
Baseline	74.61
AISPACE	**90.85**
Unibuc-NLP	86.96
dianchi	83.48
L3i++	83.12
USTC-BUPT (ours)	84.33
w/o three-stage strategy	80.94

Table 7: Part of the official results for subtask B.

System	MAE
Baseline	21.535
TM-TREK	**15.684**
AIpom	15.940
Fine-tuned RoBERTa-large	20.876
Fine-tuned DeBERTa-large	18.075
USTC-BUPT (ours)	17.702
with Bi-LSTM	16.556

Table 8: Part of the official results for subtask C.

5.2.2 Ablation Study

We conduct extensive ablation experiments to show the effectiveness of LLM Embeddings and LLM Sentinel respectively. The results are shown in Table 6. When we remove the language discriminator and only use the LLM Sentinel method, the accuracy drops to 92.03%. Since mT5 is designed for multilingual text-to-text tasks, its training corpus may be more biased towards encompassing texts in diverse languages rather than focusing on a specific language, such as English. Consequently, this could result in inferior performance on English text classification tasks due to the model's lack of exposure to a sufficient amount of English texts for optimal training. However, when we solely utilize the LLM Embeddings method, the accuracy decreases to 82.05%. Since most of the training data for Llama-2-70B is in English, its ability to comprehend other languages is limited. The potential of other multilingual LLMs awaits exploration in future research. A more detailed discussion is in the Appendix C.

5.3 Subtask B: Multi-Way Track

Our final submission achieves an accuracy of 84.33%, marking a 9.72% improvement over the baseline, as shown in Table 7. This indicates that LLM can capture subtle differences between different models, allowing for classification based on these distinctions. Furthermore, rather than directly implementing multi-classification, we embrace a three-stage strategy. This results in an enhancement in model performance from 80.94% to 84.33%, suggesting that we can prioritize the classification of categories with a significant gap before handling the others.

5.4 Subtask C: Mixed Track

After the release of the golden label, we test the performance of our model with a Bi-LSTM (Zhou et al., 2016) layer. The MAE of our final submission is 17.702, and the new result is 16.556, only slightly larger than the SOTA benchmark by TM-TREK but saliently smaller than the baseline as seen in Table 8. This shows the strong ability of DeBERTa to extract effective contextualized features, while LSTM (Hochreiter and Schmidhuber, 1997) helps process sequential information in the text. The result compared with DeBERTa-large also shows that the effectiveness of the encoder model is not linear with the scale.

6 Conclusion

In conclusion, this paper presents the development and performance of our system for the SemEval-2024 Task 8. Our system wins the multilingual track and secures second place in the monolingual track. Additionally, we attain third place in both subtask B and subtask C. We demonstrate the efficacy of incorporating DANN, which significantly enhances out-of-domain accuracy by introducing a gradient reversal layer and integrating additional domain labels. Leveraging LLM embeddings proves to be a straightforward yet effective method, harnessing the representation capabilities of LLM without fine-tuning. Furthermore, our implementation of LLM Sentinel exhibits remarkable performance, especially in low-resource language scenarios. In the future, we plan to investigate the application of DANN to multi-label classification scenarios and explore more effective strategies to leverage LLM embeddings.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.62232006, 62222212, 62121002, 62376033).

References

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *ArXiv*, abs/1906.03351.

Ning Bian, Hongyu Lin, Peilin Liu, Yaojie Lu, Chunkang Zhang, Ben He, Xianpei Han, and Le Sun. 2023. Influence of external information on large language models mirrors social cognitive patterns.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Token prediction as implicit classification to identify LLM-generated text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13112–13120, Singapore. Association for Computational Linguistics.

Zhuyun Chen, Guolin He, Jipu Li, Yixiao Liao, Konstantinos Gryllias, and Weihua Li. 2020. Domain adversarial transfer network for cross-domain fault diagnosis of rotary machinery. *IEEE Transactions on Instrumentation and Measurement*, 69(11):8702–8712.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks.

Hans W. A. Hanley and Zakir Durumeric. 2024. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites.

Mahta HassanPour Zonoozi and Vahid Seydi. 2023. A survey on adversarial domain adaptation. *Neural Processing Letters*, 55(3):2429–2469.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. 2023. Origin tracing and detecting of llms.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023. Gpt-4 technical report.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the risk of misinformation pollution with large language models.

Mike Perkins, Jasper Roe, Darius Postma, James McGaughran, and Don Hickerson. 2023. Detection of gpt-4 generated text in higher education: Combining academic judgement and software to identify generative ai tool misuse. *Journal of Academic Ethics*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, et al. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *ArXiv*, abs/1908.09203.

Edward Tian. 2023. GPTZero: An ai text detector.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Christoforos Vasilatos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. 2023. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023a. Seqxgpt: Sentence-level ai-generated text detection.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *ArXiv*, abs/2305.14902.

Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *ArXiv*, abs/2305.17359.

Haolan Zhan, Xuanli He, Qiongkai Xu, Yuxiang Wu, and Pontus Stenetorp. 2023. G3detector: General gpt-generated text detector.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212.

A Data

In the monolingual track of subtask A, the dataset contains both human-written and machine-generated text. Different from the training and development set, all the data in the test set comes from a new area: student essays Outfox, and a new generator GPT-4 appears to generate machine text.

In the multilingual track, the text is not only in English, but also in Chinese, German, Russian and other languages. Compared to the training and the development set, two new fields of Outfox and Italian text appear in the test set, and new generators Llama-2-finetune and Jais-30B (Sengupta et al., 2023) are used to generate machine text.

In the dataset of subtask B, the generators remain the same in the training set, development set and test set, including Human, ChatGPT, Cohere, Davinci, BLOOMZ and Dolly. However, the text of the test set is only from the student essays Outfox instead of wikiHow, etc.

Each text of the subtask C dataset is composed of human-written text and machine-generated text, and its label is an index, representing the boundary where the change occurs.

Tables 9 to 12 detail the data sources and the distribution of the model, which is conducive to evaluating the model's generalization ability.

	Train	Dev	Test
wikiHow	✓	✓	
Wikipedia	✓	✓	
Reddit	✓	✓	
arXiv	✓	✓	
PeerRead	✓	✓	
Outfox			✓

Table 9: Source distribution of subtask A monolingual track.

	Train	Dev	Test
Human	✓	✓	✓
ChatGPT	✓		✓
Cohere	✓		✓
Davinci	✓		✓
Dolly	✓		✓
BLOOMZ		✓	✓
GPT-4			✓

Table 10: Model distribution of subtask A monolingual track.

	Train	Dev	Test
wikiHow	✓		
Wikipedia	✓		
Reddit	✓		
arXiv	✓		
PeerRead	✓		
Bulgarian	✓		
Urdu	✓		
Indonesian	✓		
Chinese	✓		
Russian		✓	
Arabic		✓	✓
German		✓	✓
Outfox			✓
Italian			✓

Table 11: Source distribution of subtask A multilingual track.

	Train	Dev	Test
Human	✓	✓	✓
ChatGPT	✓	✓	✓
Cohere	✓		✓
Davinci	✓	✓	✓
Dolly	✓		✓
BLOOMZ	✓		✓
Llama 2			✓
Jais-30B			✓

Table 12: Model distribution of subtask A multilingual track.

B Detailed Experimental Setup

B.1 Domain-Adaptive Text Detection

Typically, the number of samples in the training set far exceeds that in the development set, which can also be observed in this competition dataset. The imbalance between the domain labels of the source and target domains is substantial, with 119,757 samples in the source domain and only 5000 samples in the target domain. To address this issue, we innovatively repeat the target domain 15 times to achieve a nearly 1:1 ratio of domain labels between the source and target domains, without compromising the genuine domain and classification label values of the target domain.

We opt to utilize the pre-trained model RoBERTa-base as the feature extraction layer for DANN to extract features from the text to be detected. Subsequently, we input the text information separately into the label classifier and the domain classifier. Apart from the batch size, which is set to 32, which differs from the baseline, all other hyperparameters remain consistent with the baseline. Specifically, the learning rate is set to 2e-5, and the optimizer selected is AdamW (Loshchilov and Hutter, 2017). The maximum token truncation length for the text is set to 512 tokens. We conduct training for 10 epochs on a single NVIDIA A40 40GB GPU.

B.2 LLM-Powered Language-Aware Model

LLM Embeddings We utilize the int8 quantized variant of Llama-2-70B as the proxy LLM for obtaining embeddings on a single NVIDIA A800 80GB GPU, with the maximum length set to 1024. For the two-stage CNN, the input channel is set to 1. A total of three convolutional layers are employed, with the number of kernels being 32, 64, 96 respectively. The sizes of their corresponding kernels are 24, 16, 8. We use the AdamW optimizer with a linear warmup decay learning schedule and a dropout of 0.1. The batch size and learning rate are set to 128 and 3e-4, and models are trained for 50 epochs.

LLM Sentinel We fine-tune the mT5-large model for 15 epochs using two NVIDIA A40 GPUs. Throughout this process, we utilize the Adafactor optimizer (Shazeer and Stern, 2018) to minimize GPU memory usage and expedite training. The optimizer utilizes the following hyperparameters: a learning rate of 1e-3, stability parameters of (1e-30, 1e-3), gradient clipping threshold of 1.0, learning rate decay rate of -0.8, momentum parameter set to None, weight decay of 0.0, relative step set to False, parameter scaling set to False, and warm-up initialization set to False. The maximum length constraint is set to 1024.

B.3 Three-stage LLM Embeddings

The hyper-parameters of this experiment are consistent with the method mentioned above for LLM embeddings. Please refer to Appendix B.2 for more details.

B.4 Augmented DeBERTa Model

All hyper-parameters synchronize with the baseline. We only change the model structure and fine-tune it using a single NVIDIA GeForce RTX 3090 GPU.

Method	Accuracy
T5-small (directly)	77.06
T5-small (prompt)	**87.76**
LLM Embeddings (English)	**97.30**
LLM Sentinel (English)	82.04
mT5-large	**90.58**
mT5-xl	73.56

Table 13: Performance comparison of different methods on the development set.

C More Analysis of Multilingual Track

C.1 Prompt Impact

In the initial stages of the experiment, we compared the impact of adding prompts on model performance for monolingual binary classification tasks. We trained and tested the T5 model using the monolingual training and development sets, respectively. The experimental results (Table 13) indicate that adding prompts could effectively enhance the model's performance on this task. Therefore, for multilingual task, we directly adopt the approach of adding prompts.

C.2 Language-Aware Strategy

During the experimental phase, we compared the performance of LLM Embeddings and LLM Sentinel on English texts. We trained them using the monolingual training set of subtask A and validated the monolingual development set. The experimental results are presented in Table 13. The results demonstrate that LLM Embeddings outperform LLM Sentinel. Consequently, for English text, we opt for LLM Embeddings.

C.3 LLM Selection

In the later stages of the experiment, we also explored larger models, such as mT5-xl. Considering that the test set in the competition does not include Russian, we evaluated the performance of mT5-xl on languages other than Russian for this task. We trained the mT5-xl model using the whole multilingual training set and utilized texts from languages other than Russian in the multilingual development set as the new development set. The training was conducted with the same experimental parameters as mT5-large (see details in Appendix B.2). We compare the best accuracy results of mT5-large and mT5-xl on the new development set (Table 13). The experimental results indicate that employing larger models with more parameters does not yield better experimental outcomes for this task. Therefore, we choose mT5-large as our LLM.

ALF at SemEval-2024 Task 9: Exploring Lateral Thinking Capabilities of LMs through Multi-task Fine-tuning

Seyed Ali Farokh
Department of Computer Engineering
Amirkabir University of Technology
alifarokh@aut.ac.ir

Hossein Zeinali
Department of Computer Engineering
Amirkabir University of Technology
hzeinali@aut.ac.ir

Abstract

Recent advancements in natural language processing (NLP) have prompted the development of sophisticated reasoning benchmarks. This paper presents our system for the SemEval 2024 Task 9 competition and also investigates the efficacy of fine-tuning language models (LMs) on BrainTeaser—a benchmark designed to evaluate NLP models' lateral thinking and creative reasoning abilities. Our experiments focus on two prominent families of pre-trained models, BERT and T5. Additionally, we explore the potential benefits of multi-task fine-tuning on commonsense reasoning datasets to enhance performance. Our top-performing model, DeBERTa-v3-large, achieves an impressive overall accuracy of 93.33%, surpassing human performance. The code and models associated with this study are publicly available at https://github.com/alifarrokh/SemEval2024-Task9.

1 Introduction

The SemEval 2024 Task 9, BrainTeaser, is a multiple-choice question-answering task, organized by (Jiang et al., 2024) and based on the BrainTeaser benchmark (Jiang et al., 2023) that aims to test the ability of NLP models to exhibit lateral thinking, a creative type of human reasoning process that often requires looking at problems from a new perspective. Unlike similar benchmarks for computational creativity, such as RiddleSense (Lin et al., 2021), which focus on problems resolvable through commonsense associations, the BrainTeaser benchmark comprises questions that challenge models to defy default commonsense associations and linear inference chains (Jiang et al., 2023).

The task includes two subtasks: Sentence Puzzle and Word Puzzle. While the puzzles in the first subtask focus on the meaning of sentences, the word puzzles concentrate on the letter composition of questions and their choices. The following are examples of questions in each subtask.

- **Example Sentence Puzzle**
  Question: A man shaves everyday, yet keeps his beard long. How is that possible? (A) He is a barber. (B) He wants to maintain his appearance. (C) He wants his girlfriend to buy him a razor. (D) None of above.
  Answer: A

- **Example Word Puzzle**
  Question: What part of London is in France? (A) The letter O. (B) The letter N. (C) The letter L. (D) None of above.
  Answer: B

(Lin et al., 2021) discusses three types of popular methods for commonsense question answering: 1) Fine-tuning pre-trained language models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), 2) Fine-tuning text-to-text question answering models such as T5 (Raffel et al., 2020), 3) Incorporating knowledge graphs for graph-based language reasoning similar to KagNet (Lin et al., 2019) and MHGRN (Feng et al., 2020). An advantage of using graph-based reasoners is the interpretability of their results due to the symbolic structures of knowledge graphs. Motivated by the superior performance achieved by fine-tuning language models or text-to-text models in achieving the best results on the RiddleSense benchmark, our study investigates the vertical thinking capabilities of these models. We accomplish this by fine-tuning them on the BrainTeaser dataset.

We solely engage in the first subtask of BrainTeaser (Sentence Puzzles) and, due to resource constraints, confine our experiments to models with fewer than one billion parameters. In the subsequent section (Section 2), we provide a brief discussion of the models we fine-tuned. Subsequently, we offer a more detailed introduction to the task in

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1523–1528

Section 3. Section 4 delves into the specifics of our experiments and their outcomes, while Section 5 presents our results in the competition alongside a concise error analysis.

2 System Overview

Inspired by the recent progress in pre-trained language models, our work investigates the performance of fine-tuned language models on the BrainTeaser task. Specifically, we fine-tuned two groups of models, i.e., BERT-based and T5-based models.

2.1 BERT-based Models

The models included in this group are ALBERT v2 (Lan et al., 2019)[1], RoBERTa (Liu et al., 2019), and DeBERTa v3 (He et al., 2023). We refer to this group as BERT-based models because all of them are inspired by BERT, a pre-trained bidirectional transformer encoder (Vaswani et al., 2017), with slight improvements in their pre-training objectives or architectures. The overall process of fine-tuning BERT-based models for multiple choice question answering is illustrated in Figure 1.

Note that for the experiments in which multiple datasets with different numbers of choices are used during fine-tuning, we have to normalize the questions so they consist of the same number of choices, and the model can be fine-tuned with a shared linear projection layer. This is simply achieved by either randomly removing extraneous options from questions with too many choices or by adding dummy options to other ones. Since dummy options are constant in all the questions, the model can easily learn to ignore them and assign a zero probability to them.

As a side note, we also fine-tuned BERT in a sequence classification format where all options are fed into the model so it can infer the correct one by looking at the others. However, the performance was suboptimal in this case, so we did not include the results in the paper.

2.2 T5-based Models

This group includes Flan T5 (Chung et al., 2022) and Unified-QA v2 (Khashabi et al., 2022), pre-trained encoder-decoder transformers that convert all NLP problems into a text-to-text format. These models are fine-tuned to generate the correct choice conditioned on the input question (Figure 2).

[1]ALBERT v2 was introduced in their GitHub repository at https://github.com/google-research/albert

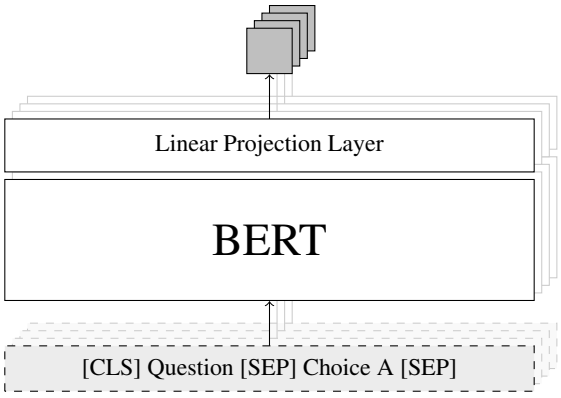


Figure 1: Fine-tuning BERT for multiple-choice question answering involves computing n forward passes simultaneously for questions with n choices. The output embeddings are then projected into a vector of size n, which is fed into a SoftMax function to compute the Cross-Entropy Loss. This optimization process aims to maximize the score of the correct choice.

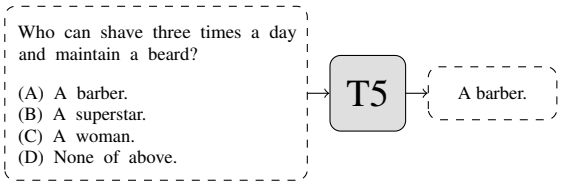


Figure 2: Fine-tuning T5-based models for multiple choice question answering.

3 Task Overview

3.1 Adversarial Examples

The BrainTeaser dataset includes two types of adversarial examples for each original data: *Semantic Reconstruction* and *Context Reconstruction*. In semantic reconstruction, the original question is rephrased so that it conveys the same meaning with the same answer. Extraneous options (i.e., other choices) are kept unchanged in this construction method. In context reconstruction, on the other hand, both the original question and choices are changed so that they describe a new situational context with the same reasoning path as the original question.

3.2 Dataset

The BrainTeaser dataset (Sentence Puzzle) consists of train and test splits, containing 169 and 40 original data along with their adversarial examples, totaling up to 507 and 120, respectively. The test set was released after the evaluation phase was over. Furthermore, a subset of the training data consisting of 102 examples was selected as the validation set during the evaluation phase. However,

Model	BS	LR
ALBERT v2 xlarge	48	1e-5
ALBERT v2 xxlarge	48	1e-5
DeBERTa v3 base	48	25e-6
DeBERTa v3 large	48	11e-6
RoBERTa base	64	1e-5
RoBERTa large	64	1e-5
Flan T5 base	24	5e-4
Flan T5 large	8	4e-4
Unified QA v2 base	24	5e-4
Unified QA v2 large	8	4e-4

Table 1: The hyper-parameters used for fine-tuning our models. LR indicates the Learning Rate and BS shows the Batch Size.

as described in Section 4.2, we chose to employ k-fold cross-validation instead of relying solely on the validation set for model development.

3.3 Evaluation Metrics

The task organizers have defined two types of accuracy metrics to evaluate the performance of models: *Instance-based accuracy*, where each question is considered a separate instance, and *Group-based accuracy*, where each question and its adversarial instances form a group and systems are given an accuracy of one only when they correctly predict all questions in the group.

We refer to the instance-based accuracy on all examples as `overall` accuracy and the instance-based accuracy on `original`/`semantic`/`context` examples as `ori`/`sem`/`con` accuracy. Correspondingly, `ori-sem` and `ori-sem-con` denote the group-based accuracy of their corresponding questions.

4 Experimental Setup and Results

4.1 Implementation Details

All models were implemented in Python using the Transformers (Wolf et al., 2020) library. AdamW (Loshchilov and Hutter, 2017) was used for optimization, and all models were fine-tuned for 4 epochs. Due to resource constraints, we only tuned the effective batch size and Learning Rate (LR) of models using grid search. See Table 1 for the list of hyper-parameters used for fine-tuning models.

Dataset(s)	# Samples	CV Accuracy
RS	3,510	**81.43**
CSQA	9,741	79.66
PIQA	16,113	79.48
SIQA	33,410	79.95
HellaSWAG	39,905	78.48
SWAG	73456	76.51
BrainTeaser		75.53

Table 2: The 5-fold cross-validation accuracies of models fine-tuned on a union of different commonsense datasets and BrainTeaser (BT), compared with the accuracy of a model fine-tuned on BrainTeaser only.

4.2 Reliability of Experiments

During the development of our models, we noticed that the limited number of training and validation examples led to noisy results when evaluating the original validation set. Consequently, relying solely on this set for model development was deemed unreliable. Therefore, we used 5-fold cross-validation to perform our experiments in the evaluation phase of the competition. Data folds were created by splitting the 169 groups into five sections, ensuring that questions from the same group would not appear in both the training and validation sets. Moreover, we observed that the random initialization of linear projection layers in BERT-based models causes significant variations in the performance of models. Therefore, we repeated the experiments related to BERT-based models three times and averaged the results to increase the reliability.

4.3 Auxiliary Datasets

In contrast to prior vertical thinking datasets, such as PIQA (Bisk et al., 2020) and RiddleSense (Lin et al., 2021), solving BrainTeaser's lateral thinking puzzles requires more creativity and defying preconceptions (Jiang et al., 2023). Our hypothesis is, however, that although combining vertical thinking datasets with BrainTeaser may not directly improve our model's performance, it can provide our model with some knowledge that might be helpful during the reasoning process. For instance, solving the example puzzle in Figure 2 requires the model to have some common sense about what barbers do and what they do not. Another reason why using auxiliary datasets during fine-tuning might be helpful is that fine-tuning large models on small datasets, such as BrainTeaser's training set, can

increase the risk of overfitting, which may be prevented by using more training data.

Some datasets that cover various aspects of commonsense reasoning are RiddleSense (RS) (Lin et al., 2021) for computational creativity, CommonSenseQA (CSQA) (Talmor et al., 2018), SWAG (Zellers et al., 2018), and HellaSWAG (Zellers et al., 2019) for general commonsense knowledge, Social IQA (SIQA) (Sap et al., 2019) for social psychology knowledge, and Physical IQA (PIQA) (Bisk et al., 2020) for physical knowledge. To determine which ones can be effective for our task, we fine-tuned a `Flan-T5-base` model on the union of BrainTeaser's training set and each of the mentioned dataset's training data, and compared their accuracies with a similar model fine-tuned on BrainTeaser only (Table 2). As expected, fine-tuning on a combination of BrainTeaser and commonsense datasets enhances the model's performance in all cases. It is also notable that, despite being the smallest dataset, RiddleSense improves the model's accuracy more than any other dataset, possibly because of its distribution overlap with BrainTeaser, as they both have been collected from public websites and deal with computational creativity.

Following (Khashabi et al., 2020), we generate training batches so that each one contains almost the same number of examples from each dataset.

The datasets mentioned in our study serve as valuable resources for enhancing the performance of our multiple-choice QA (MCQA) models. Among these datasets, RS, CSQA, and PIQA are inherently structured as MCQA datasets, making them suitable for direct use in our experiments. However, to incorporate SWAG, HellaSWAG, and SIQA into our study, we need to transform their formats into MCQA. For SWAG, we consider `sent1` as the question and concatenate `sent2` with all potential endings to create the options. Similarly, in HellaSWAG, `ctx-a` is treated as the question, while `ctx-b` is prepended to each possible ending to form the options. Finally, in SIQA, the combination of the `context` and `question` fields in each sample constructs the final question.

4.4 Model Selection

As discussed in Section 2, we fine-tuned two groups of models, BERT-based and T5-based models. Following the results of the previous section (Section 4.3), all models were fine-tuned on a combination of BrainTeaser and RiddleSense. Despite the potential performance improvement from including other datasets, we limited our training set to RiddleSense and BrainTeaser for computational feasibility.

Metric	Accuracy	Ranking
ori	92.5	4
sem	95.0	3
con	82.5	6
ori-sem	92.5	4
ori-sem-con	82.5	5
overall	90.0	7

Table 3: The accuracies and rankings of our submission based on different official metrics. Refer to Section 3.3 for more details about the evaluation metrics.

The reported results in Table 4 indicate that Unified-QA's performance is approximately on par with or outperforms Flan T5. This is expected because Unified-QA-v2 was specifically trained for question answering on many QA datasets, including CSQA, PIQA, and SIQA (Khashabi et al., 2022), which can enhance the performance on BrainTeaser as shown in the previous section. In the case of BERT-based models, not only does DeBERTa-v3 surpass all other BERT-based models, but it also achieves the highest test accuracy among all models and slightly outperforms the human performance, suggesting the effectiveness of its architecture for this task.

5 Results and Error Analysis

5.1 Competition Results

We submitted our DeBERTa-v3-large [2] model (Table 4) during the competition, ranking 7 in the official leaderboard. See Table 3 for more details.

5.2 Error Analysis

There is a 12.5% gap between the accuracies of our best DeBERTa-v3 model on `ori-sem` and `con` (see Table 5), signifying that even though our model learns the semantics of puzzles very well, it sometimes fails to generalize the underlying reasoning paths to other similar situations. This gap is much narrower (5%) for our Unified-QA-v2 model, which outperforms the DeBERTa-v3 on context-

[2]Please note that the DeBERTa-v3-large checkpoint used in our submission was selected before the release of the official test set. For analysis of our best checkpoint, refer to Section 5.2.

Model	# Params	CV Accuracy	Test Accuracy [1]
ALBERT v2 xlarge	59M	79.38	75.83
ALBERT v2 xxlarge	223M	76.06	83.33
RoBERTa base	125M	81.42	80.83
RoBERTa large	355M	83.47	86.67
DeBERTa v3 base	184M	85.90	87.50
DeBERTa v3 large [2]	434M	**89.47**	**93.33**
Flan T5 base	223M	81.43	82.50
Flan T5 large	750M	82.22	84.17
Unified QA v2 base	223M	80.49	84.17
Unified QA v2 large	734M	80.64	90.08
Human (Jiang et al., 2023)	-	-	91.98

Table 4: The overall 5-fold cross-validation and test accuracies of BERT-based and T5-based models
[1] Best accuracies on the official test set released after the evaluation phase
[2] Our submission during the evaluation phase

reconstruction adversarial examples by 2.5% despite underperforming it on original and semantic-reconstruction examples, suggesting that T5-based models may learn to generalize the reasoning paths in the BrainTeaser task better than BERT-based models.

The Unified-QA-v2 model also outperforms DeBERTa-v3 on questions to which *"None of above."* is the answer (see Table 5), which is expected because T5-based models have access to all possible choices while BERT-based models can only see one choice at a time (see Figure 1 and Figure 2).

Five of the six groups that included incorrect predictions from DeBERTa-v3 and Unified-QA-v2 (see Table 5) are identical, and among the errors made in these five groups, six out of seven wrong predictions belong to the same questions, which indicates that the two models almost made the same mistakes. Analyzing those six questions shows us that half of them are related to the models' understanding of math.

Metric	DeBERTa-v3	Unified-QA-v2
ori	97.5	92.5
sem	97.5	92.5
con	85.0	87.5
ori-sem	97.5	92.5
ori-sem-con	85.0	85.0
overall	93.3	90.8
choice d[1]	87.0	93.0
false answers	8	11
false groups	6	6

Table 5: A comparison between the performance of our best models - [1]Overall accuracy of questions to which *"None of above."* is the answer.

6 Conclusion

In this study, we investigated the effectiveness of fine-tuning various language models (including BERT-based and T5-based models) on the BrainTeaser benchmark. We demonstrated the efficacy of multi-task fine-tuning on additional commonsense datasets and its impact on performance in BrainTeaser.

Although our best models achieved performance surpassing human levels, it's important to note that our study was limited to language models with fewer than one billion parameters and training sets comprising at most two datasets combined. Future research could explore extending this study in these directions, as well as investigating other aspects of computational creativity and question-answering.

We hope that our work inspires future research in these areas and contributes to the ongoing advancement of natural language understanding and reasoning.

References

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the*

AAAI conference on artificial intelligence, volume 34, pages 7432–7439.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. *arXiv preprint arXiv:2005.00646*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Daniel Khashabi, Yeganeh Kordi, and Hannaneh Hajishirzi. 2022. UnifiedQA-v2: Stronger generalization via broader cross-format training. *arXiv preprint arXiv:2202.12359*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UnifiedQA: Crossing format boundaries with a single QA system. *arXiv preprint arXiv:2005.00700*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. RiddleSense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. *arXiv preprint arXiv:2101.00376*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. SocialIQa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. CommonSenseQA: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Pollice Verso at SemEval-2024 Task 6: The Roman Empire Strikes Back

Konstantin Kobs[†]
anacision GmbH
konstantin.kobs@anacision.de

Jan Pfister[†] and **Andreas Hotho**
Data Science Chair, CAIDAS
University of Würzburg (JMU)
{lastname}@informatik.uni-wuerzburg.de

Abstract

We present an intuitive approach for hallucination detection in LLM outputs that is modeled after how humans would go about this task. We engage several LLM "experts" to independently assess whether a response is hallucinated. For this we select recent and popular LLMs smaller than 7B parameters. By analyzing the log probabilities for tokens that signal a positive or negative judgment, we can determine the likelihood of hallucination. Additionally, we enhance the performance of our "experts" by automatically refining their prompts using the recently introduced OPRO framework. Furthermore, we ensemble the replies of the different experts in a uniform or weighted manner, which builds a quorum from the expert replies. Overall this leads to accuracy improvements of up to 10.6 p.p. compared to the challenge baseline. We show that a Zephyr 3B model is well suited for the task. Our approach can be applied in the model-agnostic and model-aware subtasks without modification and is flexible and easily extendable to related tasks.

1 Introduction

Language Models Are Outstanding, but[2] they can hallucinate, i.e. generate texts that are not supported by the input or the context. Hallucinations can undermine the credibility and usefulness of LLMs, especially for applications that require high accuracy and reliability, such as summarization, question answering, or dialogue. Therefore, there is a pressing need for developing methods to detect and mitigate hallucinations in LLMs, as well as to understand the causes and effects of this phenomenon.

In this SemEval challenge (Mickus et al., 2024), the task is to detect LLM hallucinations based on the input task given to the LLM, the LLM response, and the ground truth answer. Here, the input tasks can be *definition modeling (DM)*, *machine translation (MT)* or *paraphrase generation (PG)*. Each task contains multiple examples that are fed through an LLM and its response is classified as hallucination or not by five annotators. There are two subtasks in this challenge: In the *model-aware* subtask, access to the generating model is given, while in the *model-agnostic* subtask, the generating model is unknown. We approach both subtasks in the same way, by removing the model information from the model-aware subtask. While we are certain that access to the generating model can be beneficial, we argue that the model-agnostic setting has better transferability in practice.

As the competition baseline (which uses SelfCheckGPT by Manakul et al. (2023)), we frame the task of hallucintion detection as a "consistency checking" problem with the given information, where the goal is to check whether an LLM generation is supported by the ground truth. If the generation is not supported by the ground truth, new and thus probably false information must be present; the LLM has hallucinated its response.

For building an intuition for our approach, we imagine the same setting in the real world: A person responds to a question and our goal is to detect if this is a hallucination, i.e., the response is not supported by the truth. With this real-world setting in mind, we formulate three intuitions that we later transfer to the challenge baseline:

(I) Instead of one person, we ask multiple different experts to check the response's consistency with the truth and weight the different expert responses based on their past performance to make a final decision.

(II) Each expert gives a certainty for their response, so we can take this into account

[†] These authors contributed equally to this work.
[1] https://en.wikipedia.org/wiki/Pollice_verso
[2] https://x.com/ChrisGPotts/status/1686802492104028160

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1529–1536

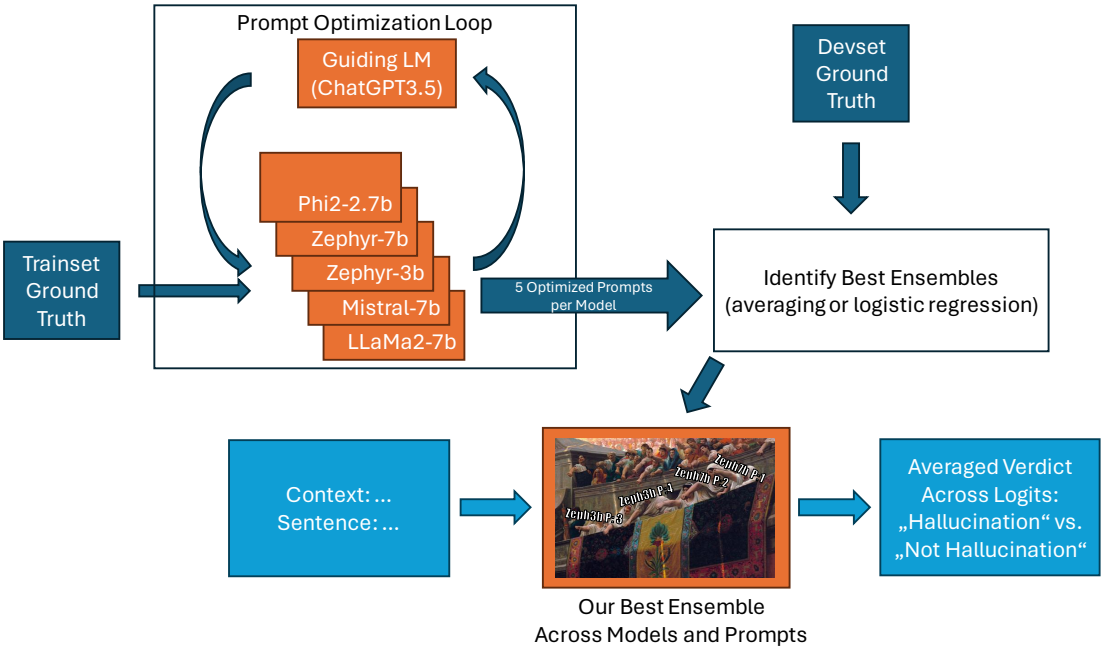


Figure 1: Overview of our approach. We automatically optimize the prompts of multiple "expert" LLMs to check the consistency of the given model output with the ground truth. We combine multiple experts (models-prompt-combinations) in a uniform or weighted manner and select the best ensemble on an internal validation set. This ensemble performs the final collective verdict, as illustrated by a part of Jean-Leon Gerome's painting "Pollice Verso" (1872), which represents the Ancient Roman gesture for judgment on defeated gladiators[1].

when combining multiple experts' responses into one final judgement.

(III) Since each expert is trained differently and has different skills, a suitable explanation of the task to the expert is beneficial.

2 System Description

Given this intuition regarding a potential approach to the task in the real world, we now connect these steps to our submitted system. An overview is found in Figure 1.

(I) We construct ensembles of multiple LLMs that are independently asked to check the response's consistency with the truth. For each ensemble of models, we combine the responses of the LLMs using averaging or by training a logistic regression (see Section 2.5).

(II) Compared to the task baseline method, we modify the procedure for how the LLMs produce their output to obtain better probability estimates (see Section 2.3).

(III) In addition to the baseline prompt, we use an automatic prompt optimization technique to create five additional, well-working prompts and use them as additional options for our ensemble selection (see Section 2.4).

2.1 Provided Baseline

In general, our system is based on the baseline provided by the task organizers. Here, a Mistral 7B model is given the following prompt:

```
Context: [GROUNDTRUTH]
Sentence: [MODELOUTPUT]
Is the Sentence supported by the context
above? Answer using ONLY yes or no:
```

The next token is then generated by the LLM and checked whether it is "yes" or "no" (possibly with additional whitespace or capitalization). If it is a "yes", the output label is set to "Not Hallucination" and its log-probability $logprob$ is converted to the probability $p(Hallucination) = 1 - e^{logprob}$. If it is a "no", the output label is set to "Hallucination" and its log-probability $logprob$ is converted to $p(Hallucination) = e^{logprob}$. If the next token is neither a "yes" or "no", the output label is chosen randomly and $p(Hallucination)$ is set to 0.5.

In the following, we apply our three intuitions to the baseline setup in order to achieve better results. To have a better understanding of which datasets

are used, we introduce them in the following section.

2.2 Data Splits

The task organizers provide multiple datasets from which we use the labeled "val.model-agnostic.json" and "val.model-aware.v2.json" files. We randomly split the model-agnostic file into two equal datasets, stratified by the task, such that both datasets have similar numbers of items with the same task.

These two datasets are our "training" and "validation" datasets, respectively. The full model-aware file is used as our internal "test" dataset, in order to estimate the performance of our system without making a submission.

The "training" dataset is used to optimize the prompts of the models. The "validation" dataset is used to train logistic regressions to weight each member of the ensemble. The "test" dataset is then used to evaluate our results internally without submitting all ensembles of models to the competition leaderboard.

2.3 Intuition (II): Better Output Generation

The task baseline from the organizers generates the most probable next token and checks if it is "yes" or "no". When running the task baseline code, we find that in 21 of the 499 examples from the "val.model-agnostic.json" file, the model does not return a "yes" or "no" directly as the highest scoring token, which means that the output label is chosen randomly and "P(Hallucination)" is set to 0.5.

We argue that it is not necessary to rely on the model to generate a "valid" token at the beginning and only hope for a definite answer. Instead, we take the models' output probability distribution over all available tokens. From this, the combined probability of relevant tokens can be accessed and computed, so even if "yes" or "no" are not the most probable tokens, a definite answer can be derived.

Let $\mathbf{T}$ be a mapping of all available LLM tokens to their corresponding log probabilities, which is accessed using $\mathbf{T}[x]$ for token x. Out of the LLM vocabulary, we identify tokens indicating a positive and negative reply, i.e. all tokens that boil down to "yes" or "no" in any capitalization and with any added whitespace. We call the sets of tokens $\mathcal{P}$ and $\mathcal{N}$ for positive and negative tokens, respectively. The probability for the answer being positive is then computed using a modified softmax function s, which takes only the positive and negative tokens into account:

$$s(\mathbf{T}) = \frac{\sum_{p \in \mathcal{P}} \exp(\mathbf{T}[p])}{\sum_{t \in (\mathcal{P} \cup \mathcal{N})} \exp(\mathbf{T}[t])} \quad (1)$$

This way, even if the token with the highest probability is not a "yes" or "no", a meaningful probability $s(\mathbf{T}) \in [0, 1]$ can be computed. This makes our system more reproducible than the organizer's baseline code, since no randomly selected labels can occur. The predicted output label is then "Not Hallucination" when $s(\mathbf{T}) \geq 0.5$ and "Hallucination" else.

2.4 Intuition (III): Prompt Optimization

We further improve the performance of our approach by "finetuning" the used prompts for each model we use in our ensemble independently. To this end, we follow the OPRO approach (Yang et al., 2023), which aims to automatically optimize the prompts with the help of a "guiding" language model. First, we take the baseline prompt as an initial starting prompt and evaluate the accuracy of the model on a split of our training dataset. Next, the guiding language model, in turn, is prompted to optimize the prompt that the model is acting upon. For this, it has access to the 20 best previously evaluated prompts, as well as the accuracy the model achieved when using this prompt. The task of the guiding language model is now to generate a new prompt that outperforms all previous prompts. Finally, the new prompt is evaluated and added to the list of tried prompts for the next optimization step.

We employ this approach to optimize the prompts for every used model separately, as the optimal prompt for one model does not have to be working well for other models. As guiding language model we select ChatGPT3.5-Turbo[3] and evaluate the prompts on our holdout set. We slightly adapt the original OPRO optimization prompt as we found the "meta" prompt submitted to the guiding language model to be very hard to decipher in our case. This stems from nested references to "below instructions" which in turn referenced the "Context above", although the samples are usually appended. As there are no clearly reserved delimiters in this prompt, this can be confusing when reading this optimization prompt — even for a human. Hence we slightly change this prompt by introducing a json-structure where fit (empty newlines have been stripped here):

[3] https://platform.openai.com/docs/models/gpt-3-5-turbo

Table 1: The best five prompts found by the prompt optimization method OPRO for the Zephyr 3B model. Overall, the optimization was run for ten iterations. Shown are also the iteration in which they were proposed and their respective accuracies on our holdout set.

Iter.	Acc.	Prompt
2	0.714	Decide whether the given sentence is directly supported by the provided context. Answer with a simple "yes" or "no".
8	0.714	Determine if the sentence provided is supported by the given context. Respond with a clear "yes" or "no".
1	0.694	Based on the given context and sentence, determine if the statement is supported or not. Please respond with a simple yes or no.
1	0.694	Based on the given context, determine if the sentence is correctly supported. Respond with a simple 'yes' or 'no'.
1	0.694	Is the Sentence consistent with the provided Context? Answer with either "yes" or "no".

```
Your task is to generate the instruction
<INS>. Below are some previous
instructions with their scores. The
score ranges from 0 to 100.
[
 {
  "<INS>": "Is the Sentence supported by
the Context above? Answer using ONLY yes
or no:",
  "score": 74
 },
 ...
]
Below are some problems commonly
solved incorrectly when using above
instructions.
[three incorrectly solved examples and
their correct label formatted as json]
Generate an instruction that is different
from all the instructions <INS> above,
and has a higher score than all the
instructions <INS> above. The instruction
should begin with <INS> and end with
</INS>. The instruction should be concise,
effective, and generally applicable to
all exemplary problems above.
```

Table 1 shows the five best performing prompts for the Zephyr 3B model found by OPRO.

2.5 Intuition (I): Ensemble Strategy

Our intuition is to ask multiple experts instead of one to assess whether the model output is hallucinated. This is implemented as an ensemble approach, where different models are asked to identify hallucinations. Their outputs are later combined to one output label and probability.

Considered Models Since there are plenty of open source language models available, we limit ourselves to five different models. We select these models from the "New & Noteworthy" section of LM Studio, a desktop application that allows to run LLMs efficiently on CPUs using quantized model weights.[4] We define several criteria to select our final models:

- In order to keep inference times low, we only consider models smaller than or equal to 7B parameters.
- To make use of the newest models, the selected models can at most be half a year old. Based on our selection date of January 10, 2024, the models have to be released (according to LM Studio) after July 10, 2023.
- We only take the newest version of a model (e.g. Mistral v0.2 is used instead of v0.1).
- We try to diversify the model architectures and training datasets by eliminating mostly Llama 2/Mistral finetuned models.
- We select general purpose LLM for the English language, i.e., no explicit code generation models or models for creative writing.

This selection process gives five models for which we download their weights in the "Q6_K"[5] quantized version: Phi 2[6], Mistral 7B Instruct v0.2[7], StableLM Zephyr 3B[8], Zephyr 7B β[9], Llama 2 7B Chat[10].

[4] The list of models can be found at https://github.com/lmstudio-ai/model-catalog/tree/205a13027c9fcd7d0c4a1874d6bb0ae45922deee/models (accessed: 2024-01-10)

[5] more information can be found here https://github.com/ggerganov/llama.cpp/pull/1684

[6] https://hf.co/TheBloke/phi-2-GGUF

[7] https://hf.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF

[8] https://hf.co/TheBloke/stablelm-zephyr-3b-GGUF

[9] https://hf.co/TheBloke/zephyr-7B-beta-GGUF

[10] https://hf.co/TheBloke/Llama-2-7B-Chat-GGUF

Table 2: Which ensembles we searched for and how we found them. Missing ones from this pattern are duplicates from other ensembles. Note the absence of LLaMas, Mistrals and Phis

				Validation		Test		Official Task Results			
								model-agnostic		model-aware	
set	**metr**	**agg**	**models**	acc	rho	acc	rho	acc	rho	acc	rho
val	acc	single	Zeph 3B P:0	0.748	0.585	0.739	0.603	0.783	0.655	0.750	0.601
val	rho	single	Zeph 3B P:4	0.736	**0.619**	0.743	**0.622**	0.776	0.687	0.747	0.597
val	acc	mean	Zeph 7B P:3 + Zeph 3B P:3	0.772	0.573	**0.766**	0.595	0.797	0.658	**0.777**	0.601
val	rho	mean	Zeph 3B P:4	0.736	**0.619**	0.743	**0.622**	0.776	0.687	0.747	0.597
val	acc	logreg	LLaMa2 7B P:3 + Mistr 7B P:0 + Zeph 3B + Zeph 7B P:0 + Zeph 7B P:2	**0.780**	0.510	0.741	0.540	0.793	0.594	0.763	0.519
val	rho	logreg	Zeph 3B P:4	0.736	**0.619**	0.743	**0.622**	0.777	0.687	0.747	0.597
test	acc	single	Zeph 7B P:2	0.708	0.490	0.749	0.482	0.746	0.525	0.743	0.418
test	rho	single	Zeph 3B	0.732	0.597	0.727	0.622	0.756	**0.690**	0.735	0.590
test	acc	mean	Zeph 3B P:3 + Zeph 3B P:4 + Zeph 7B P:1 + Zeph 7B P:2	**0.756**	0.560	**0.772**	0.585	0.799	0.646	0.772	0.560
test	rho	mean	Zeph 3B + Zeph 3B P:4	0.740	0.615	0.729	**0.626**	0.769	0.689	0.744	0.598
test	acc	logreg	Zeph 3B P:4 + Zeph 3B + Zeph 7B P:0 + Zeph 7B P:1 + Zeph 7B P:3	0.740	0.589	**0.772**	0.603	**0.803**	0.676	0.771	**0.602**
test	rho	logreg	Zeph 3B + Zeph 3B P:4	0.744	**0.617**	0.737	0.626	0.775	0.689	0.745	0.598
—	—	—	Mistral 7B (organizer's baseline)	0.644	0.338	0.695	**0.462**	0.697	0.403	0.745	0.488
—	—	—	Mistral 7B (organizer's baseline) with better output generation	**0.648**	**0.380**	**0.707**	0.452	—	—	—	—

Combining LLM Responses For each of the five models, we test overall six prompts: The baseline prompt from the organizers as well as the five best performing prompts found by OPRO. For each prompt, we compute the output labels ("Hallucination" or "Not Hallucination") and probabilities "p(Hallucination)" for the validation and test datasets. Given these 30 outputs per dataset, we combine all subsets of up to five model responses by either averaging all hallucination probabilities (*mean*) or training a logistic regression on the validation dataset (*logreg*) for a more sophisticated combination. We then evaluate all combinations on our validation and test datasets.

3 Results

We overall have three dimensions in which we can select the best model/prompt ensemble for challenge submission:

1. validation (*val*) vs. test dataset (*test*) results
2. accuracy (*acc*) vs. correlation (*rho*) as evaluation metrics
3. mean ensemble (*mean*) vs. logistic regression ensemble (*logreg*) vs. single model (*single*)

The resulting model selections as well as the organizer's baseline with their metrics for our validation and test datasets as well as the official task results are shown in Table 2. Besides the model names, if a different prompt than the baseline prompt has been used, we encode the prompt used for the model in its name ("P:0" to "P:4" with "P:0" being the automatically optimized prompt that performed best on our holdout set). We can make multiple observations in the results table.

First, the last two rows of the table show that our proposed output generation method mostly improves the baseline scores slightly. Second, all of our model ensembles are better than the organizer's baseline, showing that the ensemble and logistic regression implementations help in this setting. Note that sometimes, the best model ensemble for different dimensions is the same, e.g. Zephyr 3B P:4 is the best model when evaluated on the validation correlation, regardless of the ensemble strategy.

Third, nearly all selected model ensembles contain the Zephyr 3B model in some form, which is surprising, as the most other models have more than twice the parameters. This shows that in this task, the model size does not correlate with performance. Overall, both Zephyr (3B and 7B) models are well-suited for the task, even though they are not trained by the same companies and thus not directly related.

In terms of official task results, the model ensemble chosen by the best test accuracy (and then by correlation) shows the best accuracy of our submissions on the model-agnostic task (80.3%). It is an ensemble consisting of two versions of the Zephyr 3B model as well as three versions of the Zephyr 7B model, combined through a logistic regression. It ranks 41 out of all 260 submissions for the model-agnostic task of the challenge.

For the model-aware task, our best model ensemble again consists of a Zephyr 3B and 7B version. These models are combined using the mean ensemble strategy, leading to 0.777 accuracy as official results. This submission ranks 103 out of 295 task submissions. Given that our approach is completely model-agnostic and thus not specifically designed for this subtask, the results are fairly good.

4 Analysis

We present a comparative analysis of the predicted versus actual probabilities of hallucination across the three tasks given in the challenge datasets. To this end, we plot the predictions for the ensemble that achieved the highest accuracy on our test set (test/acc/logreg in Table 2). In Figure 2 the overall gold label distribution for the probability of hallucination (p(Hallucination)) is depicted on the left, contrasting starkly with the more extreme predicted label distribution shown in the right plot. This polarized nature of predictions suggests a tendency for our model to forecast outcomes with heightened certainty, a trait observable across all ensemble models examined.

Particularly noteworthy is the paraphrase task's label distribution, which significantly deviates from the other tasks. This unique distribution is reflected not only in the ground truth data but also in our model's predictions, indicating a consistent model response to the characteristics of this task.

Figure 3 underscores a clear positive correlation between the predicted probabilities of hallucination and the ground truth scores for all task types. Hallucination detection on the machine translation task exhibits the strongest correlation, suggesting a higher predictive performance, whereas for definition modeling and paraphrase generation the scores correlate slightly less closely.

5 Discussion and Future Work

Our proposed system is easy to understand and implement, can be applied both in model-agnostic and model-aware scenarios, and is flexible in terms of different metrics: By using smaller and fewer models in the ensemble, the runtime can be achieved. By choosing the models based on a given metric, the performance given this metric can be optimized.

In the following, we want to discuss two areas: Runtime optimization and task realism. The runtime of our system depends on the number of models. Here, we have shown that small and single models can already lead to very good results. Currently, we only use one type of quantization for all models. Exploring the effects of different quantization methods might be interesting, since usually, with higher quantization, the model size gets smaller and the model gets faster, but performance degrades. Since the Zephyr 3B model is the best model for this task, maybe another quantization can optimize the runtime of our system even further. Additionally, since most of our models contain multiple versions of a prompt for the same model, we could employ batching of these prompts. This could also reduce the runtime of the system.

Currently, our prompt that is fed into the models contains the model output as well as the ground truth. The model then is instructed to check whether the model output is grounded in the ground truth. This approach follows the baseline provided by the organizers. Consequently, our prompt optimization only uses these two inputs as well. It might be interesting to evaluate, whether using the model input as an additional prompt input can increase the performance of the system.

Overall, we argue that in a realistic use case of our system, the ground truth is not known when checking for hallucinations. Instead, the system should check whether the provided model output is a correct answer to the model input. Since our approach is very flexible, it is possible to enable this use case in our system by altering the model prompt to contain both the model input and its generated output, removing the ground truth. Then, our system could be used as a validator step after the output of a LLM, which catches hallucinated inputs or at least outputs the "p(Hallucination)" score along with the LLM output.

6 Related Work

Hallucination detection and automated prompt optimization in LLMs are both vivid research topics, which became popular with the high demand for reliable LLM applications.

Hallucination Detection We mainly follow the survey by Huang et al. (2023) who categorize hallucinations in LLMs into two main categories: Factuality Hallucination, where the model output is factually incorrect, and Faithfulness Hallucination, where the model output might be correct but does not follow the user's directives or does not take provided context into account. For both types of hallucinations, there is research to detect them. External knowledge can help with identifying Factuality Hallucinations, since the model output can be checked against verified knowledge sources. In this challenge, external knowledge in form of the ground truth answer is given. Uncertainty estimation of the model output can also help with identifying Factuality Hallucinations, since the model is usually not certain when producing wrong output. For this, some methods use access to the model to identify

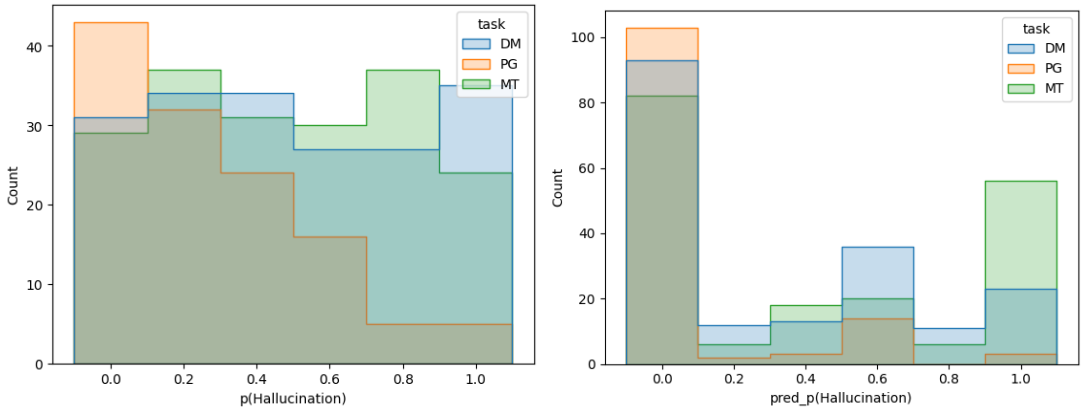


Figure 2: Gold label distributions vs. model predictions (left and right, respectively), with distinct behaviors observed for the different tasks.

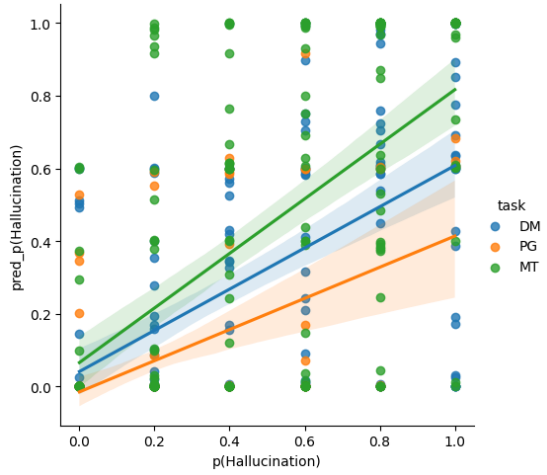


Figure 3: Positive correlation between predicted and actual hallucination probabilities for all tasks.

the uncertainty, other methods use the behavior of the model as an uncertainty indicator. The latter methods thus are model-agnostic.

For Faithfulness Hallucination, different metrics based on different methods such as word overlap or neural classifiers have been proposed. They try to identify logical misbehavior in the model output given the provided context by estimating the semantic or logical difference between the model input and its output. The idea is that Faithfulness Hallucinations stem from models not following the provided input and thus, the generated output is less consistent with its input than when the model does not hallucinate.

Prompt Optimization In the domain of prompt optimization for enhancing the reasoning capabilities of LLMs, a variety of strategies have emerged, notably in efforts to refine these models' performance through advanced prompting techniques (Qiao et al., 2023). Among these strategies, two prominent methods have recently gained attention for their novel approach to automatic prompt optimization, both leveraging a "guiding language model". This model, by having insight into the LLM's predictions, iteratively refines the prompt to achieve optimal outcomes.

The methodology introduced by Pryzant et al. (2023) draws inspiration from the principles of gradient descent and backpropagation. Initially, the guiding language model reviews the existing prompt alongside the LLM's errors, tasked with pinpointing specific shortcomings within the prompt — akin to identifying textual gradients. Following this, the same model is prompted to propose modifications that could rectify these identified issues, mirroring the process of backpropagation. This cycle of evaluation and refinement continues until the process reaches a state of "convergence".

Conversely, OPRO (Yang et al., 2023) simplifies this procedure by equipping the guiding language model with a repository of previously tested prompts and their corresponding effectiveness scores, in addition to a set of example problems. This repository provides the model with a richer context for decision-making, enabling it to discern more effectively between more and less successful prompts with each optimization iteration (as described in Section 2.4). This approach allows for a more informed and potentially more efficient refinement process.

7 Conclusion

We have introduced our system to detect hallucinations in LLMs using an ensemble strategy over multiple LLMs to decide whether the provided model output is hallucinated or not. Here, we employ a softmax over multiple relevant tokens to better capture the certainty of the models. We also employ an automatic prompt optimization scheme that finds well working prompts for each ensemble member.

We have found that the small Zephyr 3B model performs very well on this task, which motivates future exploration of its capabilities and reasons for them. Due to its simplicity, our system can be easily extended to new ensemble models and prompt templates as well as applied to new tasks, such as hallucination detection without access to ground truth.Future work might explore runtime optimizations such as batching or different model quantizations. Finally, instead of independently optimizing the model prompts, future work might jointly optimize all prompts for an ensemble, leading to different experts for different tasks.

Acknowledgements

This work is partially supported by anacision GmbH and the MOTIV research project funded by the Bavarian Research Institute for Digital Transformation (bidt), an institute of the Bavarian Academy of Sciences and Humanities. The authors are responsible for the content of this publication.

References

Jean-Léon Gérôme. 1872. Pollice verso. Oil on canvas. Phoenix Art Museum, Phoenix, Arizona. Retrieved from https://en.wikipedia.org/wiki/Pollice_Verso_(G%C3%A9r%C3%B4me).

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers.

whatdoyoumeme at SemEval-2024 Task 4: Hierarchical-Label-Aware Persuasion Detection using Translated Texts

Nishan Chaterjee [1,2,4]
[1]University of La Rochelle
La Rochelle, France

Marko Pranjić[2,4] and **Boshko Koloski**[2,4]
[2]Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Lidia Pivovarova[3]
[3] University of Helsinki
Helsinki, Finland

Senja Pollak[4]
[4]Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

In this paper, we detail the methodology of team *whatdoyoumeme* for the SemEval 2024 Task on Multilingual Persuasion Detection in Memes. We integrate hierarchical label information to refine detection capabilities, and employ a cross-lingual approach, utilizing translation to adapt the model to Macedonian, Arabic, and Bulgarian. Our methodology encompasses both the analysis of meme content and extending labels to include hierarchical structure. The effectiveness of the approach is demonstrated through improved model performance in multilingual contexts, highlighting the utility of translation-based methods and hierarchy-aware learning, over traditional baselines.

1 Introduction

Persuasion techniques in politics have a significant impact on democratic processes, which was particularly evident in contexts such as the 2020 US elections, where cognitive dissonance and media messages played a crucial role in influencing voter behaviour and attitudes (Perloff, 2013; Center, 2023). These techniques, which utilise psychological insights, align people's attitudes with their actions and thus influence political affiliations and opinions. The interplay of crises – pandemic, economic downturn, protests against racial justice, and debates over electoral legitimacy – has further highlighted the impact of persuasive narratives on public perception and democratic resilience (Jamieson et al., 2023). This complicated relationship underscores the crucial role of persuasion in political discourse and its potential to shape democratic outcomes at crucial historical moments in society.

Manually recognizing persuasion in textual content is increasingly challenging due to the vast amount of information generated daily and the nuanced nature of persuasion techniques. Efforts in this area have expanded to include the development of collaborative tasks(Da San Martino et al., 2019a) aimed at recognizing persuasion across languages and levels of hierarchy, reflecting the global and complex nature of persuasive communication in digital spaces.

In the past, researchers have used statistical text analysis methods that focused on lexical and syntactic features to identify patterns and markers of persuasive language (Jacobs, 1992). While these approaches provided basic insights, they were often not deep enough to fully capture the subtleties of human language and persuasion. The detection of persuasion in texts has shifted from statistical text analysis to the use of Large Language Models (LLMs), such as BERT (Devlin et al., 2019) and GPT (Brown et al., 2020). These models use deep learning to understand the context, semantics and complex interplay of language elements, providing more effective means of recognizing persuasive tactics in text.

The collective effort in data collection and the joint tasks have contributed significantly to belief detection, with initiatives such as the SemEval joint tasks fostering community-wide collaboration. The NLP4IF-2019 shared task(Da San Martino et al., 2019a) was another example of the collective effort to refine detection methods through standardised tasks. The task was divided into two parts: the identification of propagandistic text fragments and their specific techniques at fragment level and a binary classification at sentence level to recognise sentences containing propaganda. The joint task attracted a large participation and showed that most of the systems were able to significantly outperform the established baselines. Alhindi et al. (2019) found that for some propaganda techniques, it is not enough to look at just one sentence to make an accurate prediction (e.g. repetition) and therefore the whole article needs to be included as context. Da San Martino et al. (2019b) presented a novel method for detecting propaganda at the level of fragments in news articles that goes be-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1537–1543

yond traditional document-level detection. Their method addressed the need for more nuanced and explainable analysis by manually annotating news articles with specific propaganda techniques and developing a multi-granularity neural network model that outperformed BERT-based baselines. Koreeda et al. (2023) showed that cross-lingual and multi-task training combined with an external balanced dataset can improve genre recognition and framing, on a recently proposed task by Piskorski et al. (2023).

Recent studies have shown that translating texts from low-resource languages to a high-resource language, such as English, improves performance of end-to-end approaches on tasks such as classification (Ghafoor et al., 2021; Jauregi Unanue et al., 2023) and document similarity (Zosa et al., 2022). Koloski et al. (2023) show that cross-lingual validation can lead to improvement on classification performance for some tasks. Some earlier works also confirm this to be true for non transformer architectures (Moh and Zhang, 2012). The rest of this article is organised as follows: Section 2 describes the task, while Section 3 presents the proposed method and the results. Finally, the conclusion and proposed future work in Section 4.

2 Task description

SemEval-2023 Task 4 (Dimitrov et al., 2024) focuses on multilingual detection and classification of persuation techniques in memes.

It is composed of three subtasks. **Subtask 1** was a multi-label text classification task. The text was extracted from the image data that contained the original meme. Although the text contains less information than original image, the annotation procedure accounted for this and allows for differences between labels in the text-only data and image data that provides additional context. **Subtask 2a** was a multimodal extension of the *Subtask 1* by providing both a text and the image data. It is also a multi-label classification task, with the labels annotated based on both text and image data. **Subtask 2b** was also a multimodal task with the same inputs as *Subtask 2a*, but the task was a simpler binary classification task to detect if any persuasion technique is used.

Although only English dataset was provided for training, the evaluation was additionally done on three surprise test datasets in Bulgarian, Macedonian, and Arabic.

Dataset split	English	Bulgarian	Macedonian	Arabic
Train	7000	0	0	0
Validation	500	0	0	0
Development	1000	0	0	0
Test	1500	436	259	100

Table 1: Number of examples for each of the dataset split across languages. Notably, only English data contains data for train/val/dev split while other languages require a zero-shot approach.

We further focus only on the *Subtask 1* and its data as this was the only task we participated in.

2.1 Dataset

The input data for *Subtask 1* is the text extracted from the meme. The training, the development and the test sets were distributed as JSON files. Each of the files encoded a list of examples where each one contained text of a meme and a list of labels, together with additional metadata not used in the model (unique id of the example and URL).

Table 1 shows dataset sizes with numbers of examples in each of the dataset split for each of the languages present in the task.

Labels provided with the dataset were organized in a hierarchy that was not visible from the dataset files and the full overview of the relationships between labels was provided in the accompanying subtask description. Although 20 classes were present in the training data, their ancestors in the hierarchy provided 8 additional classes for a total of 28 classes. Submission files could provide any of the 28 classes as prediction. Predicting an ancestor class of the ground truth labels instead of the leaf-node ground truth label was counted as a partial match.

Figure 1 shows the distribution of label counts in the data. Most common labels like *Smears*, *Loaded Language*, and *Name Calling/Labeling* are almost two times more frequent than any other label. The least frequent labels like *Reductio ad hitlerum*, *Straw man*, *Red herring*, and *Obfuscation* contain less than 100 examples in train, development and validation sets combined.

2.2 Evaluation

The labels are organized in a hierarchy that can be represented as a directed acyclic graph (DAG)- a tree-like structure. Datasets presented in the shared task contained data annotated with leaf labels, but the prediction can take any of the DAG nodes: either leaf or parent. For this reason, a hierarchical

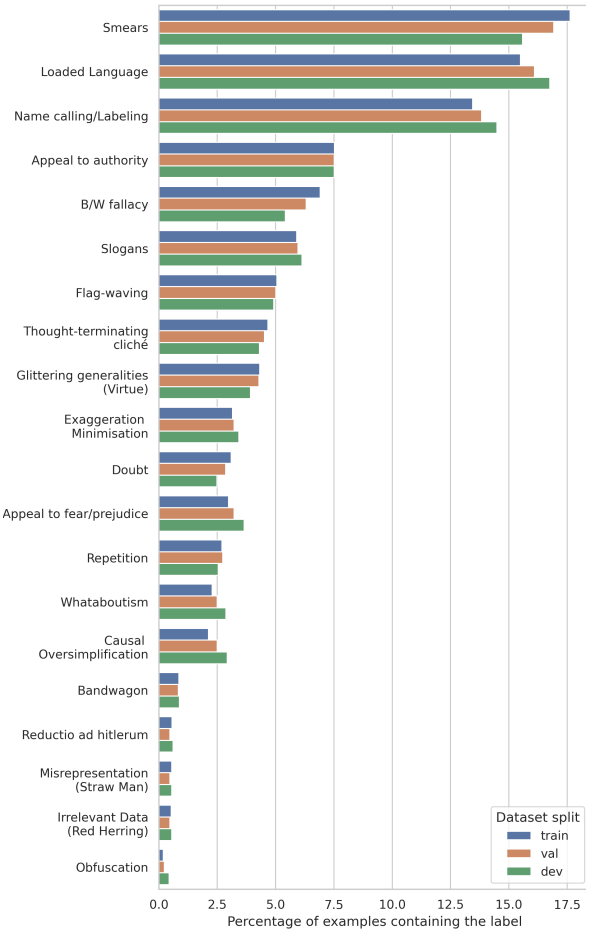


Figure 1: Label distribution shows a noticeable imbalance between class frequencies.

F_1 score (hF_1) (Kiritchenko et al., 2006) was used to take into account partially correct results, and leverage both the distance and the depth between true and predicted labels in the label hierarchy. The difference from standard, or flat, F_1 score is that the standard version considers each example as being a member of its assigned class. In contrast, the hierarchical version considers an example as a member of all parent classes in addition to its assigned leaf class. Formally, hierarchical (micro-average) version of precision (hP) and recall (hR) can be defined as:

$$hP = \frac{\sum_i |Y_A \cap \hat{Y}_A|}{\sum_i |\hat{Y}_A|} \quad hR = \frac{\sum_i |Y_A \cap \hat{Y}_A|}{\sum_i |Y_A|}$$

Where Y_A and $\hat{Y}_A$ represent a set of ground-truth and predicted labels, respectively, extended to contain all ancestors of included leaf nodes. Finally, we can define hierarchical F_1 score (hF_1) as:

$$hF_1 = 2 \cdot \frac{hP \cdot hR}{hP + hR}$$

This formulation effectively views the hierarchical classification as a multi-label setup by implicitly including hierarchy ancestor labels as additional labels.

3 Methodology and Results

Pre-trained language models, such as BERT (Devlin et al., 2018) and its variants, have shown remarkable performance across many NLP tasks. We evaluate several BERT-like models and their performance on the task. We are particularly interested in the impact of the hierarchy on model performances. We describe three approaches to understand the role of hierarchy information in the task.

First, we establish a baseline using BERT and mBART (Liu et al., 2020; Tang et al., 2020) models without using any hierarchy information and grid search to tune the hyperparameters. We explore different tokenization strategies and evaluate the model with micro-F_1.

Second, we evaluate approaches based on modifying the set of ground truth labels by extending the labels with ancestors to include hierarchy information and its influence on model performance.

Finally, we translate English train and validation data to Macedonian, Arabic, and Bulgarian with the NLLB-200 model (NLLB Team et al., 2022) and fine-tune our models for the multi-label classification task. We also compare the performance of the performance on translated data to zero-shot cross-lingual approaches using multilingual models and to translation of test sets.

3.1 Baseline Approach

We utilize distilBERT (Sanh et al., 2019) and mBERT (Devlin et al., 2018) for the baseline approach. The models are trained and evaluated without using any hierarchical information. Text data from the task was provided with escaped newlines (i.e. a newline character was represented with two characters '\' and 'n'). We evaluated a few approaches how to preprocess these data: directly tokenizing the provided text without any preprocessing (NoP), replacing the newlines with a space character (NL-Spc), or using a single newline ('\n') character (NL-L).

Both models are initialized with a linear classifier for multi-label classification. We use the AdamW optimizer (Loshchilov and Hutter, 2017) with binary cross-entropy loss and micro-F_1 score as evaluation metrics. We perform a grid search

over the learning rates (lr) [1e-4, 5e-5, 3e-5, 2e-5] and max sentence lengths of [128, 256] using a batch size of 16 over 6 epochs since BERT-based fine-tuning typically leads to decreased micro-F_1 and hF_1 scores after 6 training epochs (Sanh et al., 2019)

Results consistently favoured *lr*s of 5e-5 and 3e-5, with the highest micro-F_1 scores at lr 3e-5 with max length 128 (63.6%) and lr 5e-05 and max length 256 (63.1%) on the English validation set. However, the hF_1 score dropped to 56.9 on the English development set with the best model (see Table 2). Furthermore, we see a drop in the hF_1 on the development set when replacing the escaped newline with whitespace Table 2). We henceforth avoid any preprocessing of the input text.

We additionally compute the performance of mBART-50 (same hyperparameters) without using hierarchical information, where it gets a low mF_1 score on the validation set, but it outperforms the BERT-based models on the development set on hF_1.

Model	**Val** mF_1	**Val** hF_1	**Dev** hF_1
Performance without hierarchical information			
distilBERT (NoP)	63.6	/	56.9
distilBERT (NL-Spc)	63.6	/	50.4
distilBERT (NL-L)	63.6	/	50.4
mBERT (NoP)	59.2	/	/
mBART-50 (NoP)	48.9	59.9	59.9
Performance with hierarchical information			
distilBERT	/	60.1	**61.5**
mBART-25	/	59.8	60.4
mBART-50	**/**	**61.2**	61.0

Table 2: Performance comparison of baseline models on English dataset. Text preprocessing approach is in parentheses - no preprocessing (NoP), concatenating the lines with a single space character (NL-Spc) or using newline-separated lines (NL-L). mF_1 represents the micro F_1 score, and hF_1 indicates the hierarchical F_1 score. mBART-50 was our final submission during the official test phase.

3.2 Hierarchical Label Encoding

To include the hierarchical structure of the labels, we use the persuasion hierarchy digraph to expand the list of target labels such that it also contains all ancestor nodes. For example, [*Loaded Language*, *Name calling/Labeling*] is extended into [*Pathos*, *Loaded Language*, *Ethos*, *Ad Hominem*, *Name calling/Labeling*]. Since some labels have multiple parents, we consider all possible ancestors, and therefore [*Bandwagon*] gets extended into [*Logos*, *Justification*, *Bandwagon*, *Ethos*]. We compare the results of distilBERT and mBART-50 models compared to the approach without hierarchical label encoding. Additionally, we compute the results for mBART-25.

We internally test two approaches, a) one that extends labels for all training and validation examples and b) the one that extends the labels for training examples but not the validation examples. As extending the labels with ancestors only for the training examples consistently leads to better results, we proceed with this version.

As reported in Table 2, when using hierarchical information (extending the labels with ancestors for the training examples), mBART-25 achieves a hF_1 score of 60.4 and mBART-large-50 achieves 61.0 hF_1, on the English development set. The hyperparameters for these specific models had a learning rate of 5e-05, input size of up-to 128 tokens, and a batch size of 64.

Our results show that the hierarchical label encoding strategy consistently leads to performance improvements in this task compared to our baseline approach without hierarchical encoding, as showcased by distilBERT and mBART-50 models (see Table 2 for development set results). Note that distilBERT with hierarchy encoding results were computed in post-evaluation phase.

We submitted mBART-50 results (as it achieved the highest score on the validation set from the models we tested) for official test set evaluation for English. The model achieved 61.7 hF_1 score. We show the performance of the final model on all different language combinations in Table 4.

3.3 Translation and Test Set Results

For the three surprise test sets, we use the NLLB-200's 3.3B model (NLLB Team et al., 2022) to translate the English train and English validation sets into each target language to mimic the test stage scenario. We evaluate three settings (on the English validation sets, see Table 3): training on English and validating on translated data, training on translated data and validating on English data, and training and validating on translated data. We use the same hyperparameter grid search over the learning rates of [3e-5, 5e-5] and max lengths of [128, 256] to produce models fine-tuned for each language. Results indicate that training and validating with both target language translated sets consistently yielded better results when compared

to the other two settings (see Table 3). We use these fine-tuned checkpoints to infer the final submissions achieving results shown in Table 4.

Train	Validation	hF_1 score
EN_{train}	EN_{val}	61.2
$EN_{train} \rightarrow BG$	EN_{val}	54.9
$EN_{train} \rightarrow BG$	**$EN_{val} \rightarrow BG$**	**55.5**
EN_{train}	$EN_{val} \rightarrow BG$	47.1
$EN_{train} \rightarrow MK$	EN_{val}	53.3
$EN_{train} \rightarrow MK$	**$EN_{val} \rightarrow MK$**	**56.5**
EN_{train}	$EN_{val} \rightarrow MK$	51.3
$EN_{train} \rightarrow AR$	EN_{val}	56.2
$EN_{train} \rightarrow AR$	**$EN_{val} \rightarrow AR$**	**56.4**
EN_{train}	$EN_{val} \rightarrow AR$	50.6

Table 3: Evaluating the influence of translation as a strategy for handling low-resource languages. We measure mBART-50 model performance on Bulgarian (BG), Macedonian (MK) and Arabic (AR) by training on the English (EN) dataset translated to the target language using NLLB-200. The model is trained and evaluated on English data, possibly translated to the target language (translated dataset is shown with $\rightarrow$ followed by a target language).

Train	Validation	Test	hF_1 score
Final score on test data			
EN_{train}	EN_{val}	EN_{test}	61.7
$EN_{train} \rightarrow BG$	$EN_{val} \rightarrow BG$	BG_{test}	47.3
$EN_{train} \rightarrow MK$	$EN_{val} \rightarrow MK$	MK_{test}	36.2
$EN_{train} \rightarrow AR$	$EN_{val} \rightarrow AR$	AR_{test}	42.4
Zero-shot performance of the model			
EN_{train}	EN_{val}	BG_{test}	44.2
EN_{train}	EN_{val}	$BG_{test} \rightarrow EN$	44.2
EN_{train}	EN_{val}	MK_{test}	38.4
EN_{train}	EN_{val}	$MK_{test} \rightarrow EN$	33.8
EN_{train}	EN_{val}	AR_{test}	37.8
EN_{train}	EN_{val}	$AR_{test} \rightarrow EN$	36.4

Table 4: Evaluation of the model performance on the final test set. We measure mBART-50 model performance on Bulgarian (BG), Macedonian (MK) and Arabic (AR) by training on the English (EN) dataset. The model is trained and evaluated on English data translated to the target language (translated dataset is shown with $\rightarrow$ followed by a target language). We include final scores on test data achieved by the best-performing translation configuration. We additionally provide post-evaluation zero-shot performances of the model trained on the English data on the target language and the target language translated to English.

Using the approach where the model is trained on both train and validation data translated from English, we notice a significant degradation of the hF_1 scores for the three surprise test sets. The model for English did not use any translated data during training and validation and, as can be seen by comparing scores in Table 3 and Table 4, did not show signs of a similar degradation in performance on the test set.

This can be attributed to the distribution shift of persuasion label categories across the four languages. Our error analysis measuring accuracy for each label shows that the Arabic, Macedonian, and Bulgarian language models work well in identifying smaller classes with high accuracy while failing to generalize to the larger classes (see Table 5)[1]. These may arise from the model/training recipe failing to generalize over specific labels since different languages express persuasion strategies differently, and translations failing to capture some of these nuances.

Additionally, for our zero-shot performance evaluation, we use the model trained on English train and validation data to either directly predict the test datasets, or translating the test datasets to English and then predict. Here, we see that our translation approach works generally better than the zero-shot, except for the Macedonian dataset. This could be due to random seeds, however, larger comparable/parallel corpora are required to investigate this phenomenon.

Our final ranking on the SemEval Test Set Leaderboard are as follows: 17/32 for English, 8/20 for Bulgarian, 15/20 for Macedonian, and 4/17 for Arabic.

4 Conclusion and future work

In this paper, we describe the methods and models used by the *whatdoyoumeme* team in SemEval 2024 Subtask 1 to detect multilingual persuasion in memes. We combined two different approaches to solve this task: 1) machine translation, where we used the NLLB model (NLLB Team et al., 2022) to translate articles from English into the target languages and vice versa, and 2) including hierarchy information, where we extend a set of provided labels with labels corresponding to the ancestors nodes from the hierarchy DAG. We find that with the two proposed strategies, we can outperform both traditional encoder and decoder models, which emphasizes the importance of translation for downstream cross-lingual tasks.

In the future, we would like to extend our work

[1]All models generalize well over *Appeal to fear/prejudice* and *Distraction* but fail to generalize well over *Name calling/Labeling*.

in a few different directions. We would like to explore ensemble modelling techniques by building separate models for each belief category and using their joint predictions to improve overall performance. In addition, we would like to investigate the effects of translation quality and model size on the performance of this task.

Model	Label	Acc	Supp	Freq
mBART-50 Eng. Dev	Appeal to authority	95%	136	13.6%
	Repetition	94%	46	4.6%
	Distraction	92%	72	7.2%
	Simplification	79%	215	21.5%
	Name calling/Labeling	79%	262	26.2%
mBART-50 Arb. Test	Appeal to fear/prejudice	92%	8	8.0%
	Exaggeration/Minimisation	82%	18	18.0%
	Justification	79%	11	11.0%
	Name calling/Labeling	73%	26	26.0%
	Loaded Language	61%	24	24.0%
mBART-50 Mac. Test	Appeal to fear/prejudice	95%	13	5.02%
	Distraction	95%	11	4.25%
	Simplification	90%	10	3.86%
	Name calling/Labeling	63%	83	32.05%
	Loaded Language	61%	110	42.47%
mBART-50 Bul. Test	Appeal to authority	97%	18	4.13%
	Flag-waving	93%	28	6.42%
	Name calling/Labeling	68%	140	32.11%

Table 5: Error analysis of the model performance. Classes with higher accuracy are highlighted in green, while classes with lower accuracy in red. Only a selection of classes is shown, but a similar trend exists across all classes. Accuracy (*Acc*) is calculated using the standard binary accuracy measure. Support (*Supp*) is the number of instances from the dataset (*Train/Val/Dev/Test*) where the labels occur, and where the labels have been extended to include all ancestor nodes. Frequency (*Freq*) is calculated as support divided by the length of the dataset.

Acknowledgements

The authors acknowledge the financial support from the Slovenian Research and Innovation Agency through core research programme Knowledge technologies (No. P2-0103) and research projects: Embeddings-based techniques for Media Monitoring Applications (No. L2-50070) and Hate speech in contemporary conceptualizations of nationalism, racism, gender and migration (No. J5-3102). A Young Researcher Grant (No. PR-12394) supported the work of BK.

References

Tariq Alhindi, Jonas Pfeiffer, and Smaranda Muresan. 2019. Fine-tuned neural models for propaganda detection at the sentence and fragment levels. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 98–102, Hong Kong, China. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Annenberg Public Policy Center. 2023. Democracy amid crises: How polarization, pandemic, protests, and persuasion shaped the 2020 election.

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019a. Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 162–170, Hong Kong, China. Association for Computational Linguistics.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019b. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Abdul Ghafoor, Ali Shariq Imran, Sher Muhammad Daudpota, Zenun Kastrati, Rakhi Batra, Mudasir Ahmad Wani, et al. 2021. The impact of translating resource-rich datasets to low-resource languages through multi-lingual text processing. *IEEE Access*, 9:124478–124490.

Paul S Jacobs. 1992. Joining statistics with nlp for text categorization. In *Third Conference on Applied Natural Language Processing*, pages 178–185.

Kathleen Hall Jamieson, Matthew Levendusky, Josh Pasek, R Lance Holbert, Andrew Renninger, Yotam Ophir, Dror Walter, Bruce Hardy, Kate Kenski, Ken Winneg, et al. 2023. Democracy amid crises: Polarization, pandemic, protests, and persuasion.

Inigo Jauregi Unanue, Gholamreza Haffari, and Massimo Piccardi. 2023. T3l: Translate-and-test transfer learning for cross-lingual text classification. *Transactions of the Association for Computational Linguistics*.

Svetlana Kiritchenko, Richard Nock, and Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. volume 4013, pages 395–406.

Boshko Koloski, Blaž Škrlj, Marko Robnik-Šikonja, and Senja Pollak. 2023. Measuring catastrophic forgetting in cross-lingual transfer paradigms: Exploring tuning strategies. *arXiv preprint arXiv:2309.06089*.

Yuta Koreeda, Ken-ichi Yokote, Hiroaki Ozaki, Atsuki Yamaguchi, Masaya Tsunokake, and Yasuhiro Sogawa. 2023. Hitachi at SemEval-2023 task 3: Exploring cross-lingual multi-task strategies for genre and framing detection in online news. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1702–1711, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Teng-Sheng Moh and Zhang Zhang. 2012. Cross-lingual text classification with model translation and document translation. In *Proceedings of the 50th Annual Southeast Regional Conference*, pages 71–76.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Richard M Perloff. 2013. Political persuasion. *The SAGE handbook of persuasion: Developments in theory and practice*, pages 258–77.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Elaine Zosa, Emanuela Boroş, Boshko Koloski, and Lidia Pivovarova. 2022. EMBEDDIA at SemEval-2022 task 8: Investigating sentence, image, and knowledge graph representations for multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1107–1113.

LomonosovMSU at SemEval-2024 Task 4: Comparing LLMs and embedder models to identifying propaganda techniques in the content of memes in English for subtasks №1, №2a, and №2b

Gleb Skiba[1,2] **Mikhail Pukemo**[2] **Dmitry Melikhov**[2] **Konstantin Vorontsov**[1,2]
Institute of AI, Lomonosov Moscow State University[1],
The faculty of Computational Mathematics and Cybernetics,
Lomonosov Moscow State University[2]
gleb-skiba@mail.ru, iMan.assistance@gmail.com,
melikhov.dmitry.a@gmail.com, Voron@mlsa-iai.ru

Abstract

This paper presents the solution of the LomonosovMSU team for the SemEval-2024 Task 4 "Multilingual Detection of Persuasion Techniques in Memes" competition for the English language task. During the task solving process, generative and BERT-like (training classifiers on top of embedder models) approaches were tested for subtask №1, as well as an BERT-like approach on top of multimodal embedder models for subtasks №2a/№2b. The models were trained using datasets provided by the competition organizers, enriched with filtered datasets from previous SemEval competitions. The following results were achieved: 18th place for subtask №1, 9th place for subtask №2a, and 11th place for subtask №2b. The code for the solutions is available at github[1].

1 Introduction

In the modern world, memes are one of the most popular forms of delivering information to social media users. Unfortunately, memes created using a variety of rhetorical and psychological techniques are also used to conduct disinformation campaigns.

The overall goal of the SemEval-2024 Task 4 competition is to build models to detect rhetorical and psychological propaganda techniques in memes. The competition itself consists of three subtasks:

1. Build a model to detect rhetorical and psychological techniques only in the textual content of the meme. This is a hierarchical multilabel classification problem.

2a. Build a model to detect rhetorical and psychological techniques in both textual and visual contexts of the meme (multimodal task). This is a hierarchical multilabel classification problem.

2b. Build a model to identify the presence of rhetorical and psychological techniques in both textual and visual contexts of the meme in general. This is a binary classification problem.

In this work, experiments were conducted with generative and BERT-like models to solve subtask №1 and classifiers on top of multimodal models to solve subtasks №2a/№2b. BERT-like approaches refer to the creation classifiers over embedder models and will be used further in this article. All tasks were solved for datasets in English. The following results were achieved: 18th place for subtask №1, 9th place for subtask №2a, and 11th place for subtask №2b. The code for the solutions is available in the repository at GitHub[2].

2 Related Works

Supervised fine-tuning (SFT) (Ouyang et al., 2022) is one of the most popular methods for fine-tuning LLMs to solve various tasks. In this work, we conducted fine-tuning of LLMs, such as LLAMA[3] and Mistral (Jiang et al., 2023), for detecting propaganda techniques. Unfortunately, SFT is very resource-intensive, and conducting frequent experiments with fine-tuning LLMs is time-consuming and expensive. To address this problem, a less resource-intensive approach, LoRA (Hu et al., 2021), was used, but unfortunately, the results of this approach were not satisfactory.

In parallel with SFT and LoRA approaches, experiments were conducted on fine-tuning simple classifier layers on top of embedding models: debert (He et al., 2021), CLIP (Radford et al., 2021), BLIP (Li et al., 2022). The results of these experiments yielded comparable results to experiments with LLMs.

[1] https://github.com/pansershrek/Semeval2024_LomonosovMSU

[2] https://github.com/pansershrek/Semeval2024_LomonosovMSU

[3] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1544–1548

3 Tasks solutions

3.1 Subtask №1

This subtask represents a hierarchical classification task. Two approaches were used to solve this task:

1. Generative approach: This approach involves training a generative model to generate explicit responses to questions in JSON format.

2. BERT-like approach: This approach involves training a simple fully connected network on top of a frozen pre-trained embedding model to solve the hierarchical classification task.

3.1.1 Generative approach

Idea. The main idea of this approach is to train a generative model, both in SFT mode (Ouyang et al., 2022) and trained using the LoRA technique (Piskorski et al., 2023), to answer the question: "Does the provided text contain any propaganda techniques, and if so, which ones?"

Dataset. The following combinations of data were used for training the models:

1. For selecting the best candidate model for the final solution, the training dataset provided by the authors was used, along with the dataset from the previous semeval competition [4]. This dataset was filtered, and only samples containing propaganda techniques matching those from the current competition were taken.

2. For training the model for the final solution, the same dataset described earlier was used, along with all samples from the gold dataset added to it.

Data Format. The following prompt phrase was used: "Your goal is to identify rhetorical and psychological techniques in the given text." The model was required to output JSON with propaganda techniques or an empty JSON. The model received texts in the following format: "Your goal is to identify rhetorical and psychological techniques in the given text.\nInput:. . . \nOutput:".

All texts from the dataset were filtered as follows:

1. All unnecessary line breaks in the texts were removed.

2. All unnecessary "\" characters in line breaks were removed. That is, if the text contained a construction of the following form "\\n", it was replaced with "\n".

3. Texts with prompts and responses longer than 4096 characters were not included in the dataset.

Models. LLaMA-2-7b-chat [5] and Mistral-7b-Instruct-v0.2 (Jiang et al., 2023) were used as models.

Training and Inference Parameters. Both models were trained in both SFT mode and using the LoRA technique, using the huggingface transformers [6] and torch frameworks on 4 A100 GPUs for no more than 12 hours per experiment.

In SFT mode, both models were trained with the parameters, presented in Appendix A.1.

When training with the LoRA technique, the models were trained with the same parameters, but with the following LoRA parameters, presented in Appendix A.2.

For inference, the vLLM framework (Kwon et al., 2023) was used with the following parameters presented in Appendix A.3.

3.1.2 BERT-like approach

As a result of training with deberta-base-cased, comparable results to the generative approach were achieved: F1 score of 0.638.

Idea. The main idea of this approach is to train a fully connected layer for hierarchical classification on top of a frozen embedding model for the task of hierarchical multilabel classification.

Dataset. For model training, we utilized the training dataset provided by the authors. The dev dataset was used for selecting the final model. For the final prediction, the model was trained on a mixture of the train and dev datasets.

Data Format. To predict propaganda techniques, we represented the data labels in the format of an acyclic graph, where the nodes of this graph are generalized technique types ("Ethos", "Pathos", ..., etc.), and the leaves are concrete techniques ("Name calling", "Doubt", ..., etc.). Instead of predicting only specific techniques, we predict generalized techniques as well. For example, if the current sample needs to predict the technique {"Whataboutism"}, the model should predict Anc(y) = {"Whataboutism", "Distraction", "Reasoning", "Ad Hominem", "Ethos", "Logos", "Persuasion"}. Thus, datasets are formed as sets of pairs: (x, Anc(y)), where x is the text, and y is the set of techniques contained in the text x.

Model. The frozen model used to obtain embeddings was deberta-base-cased (He et al., 2021).

[4] https://propaganda.math.unipd.it/semeval2023task3/

[5] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[6] https://huggingface.co/docs/transformers/index

```
{
"id": "train_71410",
"input": "Your goal is to identify rhetorical and psychological techniques in the
     given text.\nInput:Tunnel to Schiff\nBunker where our next President will be
     chosen.\nOutput:",
"output": [
        "Causal Oversimplification",
        "Smears"
    ],
"full_input": "Your goal is to identify rhetorical and psychological techniques
    in the given text.\nInput:Tunnel to Schiff\nBunker where our next President
    will be chosen.\nOutput:[\"Causal Oversimplification\", \"Smears\"]"
}
```

Figure 1: Sample from the dataset example

The embedding of the entire text from the embedding model was taken, followed by the application of several dropout layers in parallel, and the results were averaged. At the end, trainable linear layers were used for classification.

Training Parameters. Training parameters of the discussed models can be seen in Appendix A.4.

3.1.3 Results

Generative approach. Experiments with pretraining models using the LoRA technique did not yield the desired result. Consistent responses from the models in JSON format could not be achieved, and there was not enough time to develop rules for formatting their outputs.

The results of the models trained with SFT are presented in the Table 1.

Mistral	LLaMa 2
0.56 F1	0.65 F1

Table 1: Results for subtask №1

BERT-like approach. As a result of training with deberta-base-cased, comparable results to the generative approach were achieved: F1 score of 0.638.

Final Prediction. For the final prediction, the LLaMA-2-7-chat model was fine-tuned on dataset 2 with the same parameters as dataset 1, and with the following parameters, shown in Appendix A.5.

It achieved an F1 score of 0.61339 and secured the 18th position on the leaderboard.

3.2 Subtask №2a

Idea. This task resembles subtask №1, but besides text, it involves meme images. It was tackled using a trainable linear layer for hierarchical classification atop frozen multimodal text-to-image embedding models.

Dataset. For model training, we utilized the training dataset provided by the authors. The dev dataset was used for selecting the final model. For the final prediction, the model was trained on a mixture of train and dev datasets.

Data Format. To represent propaganda techniques, we employed the format of an acyclic graph from the first subtask. The dataset is presented as triples: (img, x, Anc(y)), where img and x represent the image and text of the current meme, and y is the set of techniques contained in the current meme.

Models. CLIP (Radford et al., 2021) and BLIP (Li et al., 2023) were used as models to obtain embeddings for texts and images. The embeddings of the full text and the entire image were concatenated, and several dropout layers were applied in parallel, with the results averaged. At the end, trainable linear layers were used for classification.

Training Parameters. The models were trained on a single P100 GPU on the Kaggle platform. Parameters for BLIP/CLIP were frozen.

Other parameters are shown in Appendix A.6.

Results. The results of the models trained on dataset 1 are presented in the Table 2.

CLIP	BLIP
0.648 F1	0.633 F1

Table 2: Results for subtask №2a

3.3 Subtask №2b

This subtask differs from subtask №1 only in that it requires predicting whether there is any propaganda technique present in the text at all. We used the same approach, the same models with the same

hyperparameters, trained on the same datasets as in subtask №1. The only difference is in the data format - the dataset consists of triples (img, x, y), where img and x are the image and text of the current meme, and y is a flag indicating whether the current meme contains a propaganda technique.

Results. The results of the models trained on dataset 1 are presented in the Table 3.

CLIP	BLIP
0.72 F1	0.748 F1

Table 3: Results for subtask №2b

For the final prediction, the CLIP model was selected, achieving an F1 score of 0.772 and securing the 11th position on the leaderboard.

4 Further research

We did not have time to take more powerful models and experiment with them to solve subtask №1, but we are confident that Mixtral 8x7b or Llama-2-13b-chat would yield better results. Additionally, we did not have time to add data from the PTC (He et al., 2021) corpus to the dataset, but we are sure that it would have provided an even greater improvement. We also did not have time to test generative models that take text with images as input and generate text responses, for example, RUDOLPH (Radford et al., 2021).

References

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Jakub Piskorski, Nicolas Stefanovitch, Valerie-Anne Bausier, Nicolo Faggiani, Jens Linge, Sopho Kharazi, Nikolaos Nikolaidis, Giulia Teodori, Bertrand De Longueville, Brian Doherty, Jason Gonin, Camelia Ignat, Bonka Kotseva, Eleonora Mantica, Lorena Marcaletti, Enrico Rossi, Alessio Spadaro, Marco Verile, Giovanni Da San Martino, Firoj Alam, and Preslav Nakov. 2023. News categorization, framing and persuasion techniques: Annotation guidelines. Technical report, European Commission Joint Research Centre, Ispra (Italy).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

A Appendix

This appendix shows the training and generation parameters for the models described above in the text.

A.1 Hyperparameters for models training in SFT mode

- BATCH_SIZE = 4
- GRADIENT_ACCUMULATION = 4
- LEARNING_RATE = 1e-5
- MAX_LEN = 4
- WARMUP_STEPS = 4
- in fp32 and for 2 epochs

A.2 Hyperparameters for models training with LoRA technique

- LORA_R = 8
- LORA_ALPHA = 16
- LORA_DROPOUT = 0.05
- TARGET_MODULES = ["q_proj", "k_proj", "v_proj", "o_proj"]

A.3 Hyperparameters for models output generation with vLLM

- TOP_K = 50
- TOP_P = 1 and 0.9
- MAX_TOKENS = 600
- TEMPERATURE = 1, 0.8, 0.6, and 0.2

A.4 Hyperparameters for BERT-like models training for subtask №1

- Models were trained on a single P100 GPU on the Kaggle platform [7].
- Optimizer: AdamW with linear scheduler
- Learning rate (LR): 2e-5
- Batch size: 8
- Warmup steps: 100

A.5 Hyperparameters for models output generation with vLLM for final prediction

- TOP_K = 50, TOP_P = 0.9
- MAX_TOKENS = 600
- TEMPERATURE = 0.8

A.6 Hyperparameters for BERT-like models training for subtask №2a and №2b

- Optimizer: Adam
- Learning rate (LR): 2e-3
- Batch size: 10

[7] https://www.kaggle.com/

AILS-NTUA at SemEval-2024 Task 6: Efficient model tuning for hallucination detection and analysis

Natalia Griogoriadou, Maria Lymperaiou, Giorgos Filandrianos, Giorgos Stamou
School of Electrical and Computer Engineering, AILS Laboratory
National Technical University of Athens
natalygrigoriadi@gmail.com, {marialymp, geofila}@islab.ntua.gr
gstam@cs.ntua.gr

Abstract

In this paper, we present our team's submissions for SemEval-2024 Task-6 - SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes. The participants were asked to perform binary classification to identify cases of fluent overgeneration hallucinations. Our experimentation included fine-tuning a pre-trained model on hallucination detection and a Natural Language Inference (NLI) model. The most successful strategy involved creating an ensemble of these models, resulting in accuracy rates of 77.8% and 79.9% on model-agnostic and model-aware datasets respectively, outperforming the organizers' baseline and achieving notable results when contrasted with the top-performing results in the competition, which reported accuracies of 84.7% and 81.3% correspondingly.

1 Introduction

In the era that Large Language Models (LLMs) dominate and shape the trends in the Natural Language Processing (NLP) community, ensuring reliance and accurate functionality of related systems becomes a major concern. Hallucinations of language models have recently received lots of attention (Rawte et al., 2023; Ji et al., 2023; Huang et al., 2023; Ye et al., 2023; Zhang et al., 2023), questioning the trust that humans can pose in highly intelligent yet probabilistic models. At the same time, recent endeavors formally prove that hallucinations are inherent to LLMs and thus inevitable in practice (Xu et al., 2024).

Encompassing the need for detecting and analyzing hallucinations in Natural Language Generation (NLG) tasks, and given the scarcity of related datasets and benchmarks (Li et al., 2023; Cao et al., 2023; Chen et al., 2023; Muhlgay et al., 2024), the SemEval-2024 Task 6 (SHROOM: a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes) (Mickus et al., 2024) addresses the presence of semantically unrelated generations with respect to a given input, covering challenging NLP tasks, such as Machine Translation, Definition Modelling and Paraphrase Generation, which are tested both when the underlying model is known or not.

To this end, we explore efficient and widely adaptable hallucination detection strategies, tailored to the black-box demands of the problem[1]. Based on pre-trained models which contain knowledge regarding semantic relationships related to hallucinations, we achieve ∼80% accuracy in hallucination detection by fine-tuning on labeled SHROOM instances, notably higher than the 74.5% baseline accuracy provided, using an open-source Mistral instruction-tuned model[2]. Specifically, we contribute to the following:

1. We fine-tune models pre-trained on hallucination detection and Natural Language Inference (NLI) datasets, which are semantically related to SHROOM challenges.

2. Tuned models constitute a Voting Classifier, achieving competitive detection accuracy.

3. All our experimentation is time and computationally efficient, while entirely black-box.

4. Decomposition of results per task and analysis of failed and accurately detected instances provide valuable insights into the nature of the involved hallucinations.

Our code is available on GitHub [3].

[1] Even in the model-aware setting of SHROOM, we do not re-generate the outputs using the given models, therefore we continue operating in a completely black-box setup.

[2] https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF

[3] https://github.com/ngregoriade/Semeval2024-Shroom.git

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1549–1560

2 Related Work

NLP hallucinations is a rapidly evolving field, examining invalid generations from varying perspectives. Categorizations of hallucinations may view hallucinatory outputs as unfaithful to the input, inconsistent with the generated output itself, or conflicting with real-world knowledge (Zhang et al., 2023). Factual hallucinations have gathered the majority of recent breakthroughs, since comparison with existing factual sources (Lin et al., 2022; Lee et al., 2023; Chen et al., 2023; Min et al., 2023; Cao et al., 2023; Muhlgay et al., 2024) renders them accurately detectable and correctable (Chern et al., 2023; Dhuliawala et al., 2023; Li et al., 2024). The more subtle characteristics of other hallucination types constitute the creation of related benchmarks harder, not to mention techniques for automatic evaluation (Azaria and Mitchell, 2023; Kadavath et al., 2022; Manakul et al., 2023; Duan et al., 2024). A limitation tied with such techniques is that in most cases at least model probing is needed, rendering them unusable in cases where the model that produced the reported hallucinations is completely unknown or inaccessible. SHROOM comes to fill this gap, focusing on semantic faithfulness rather than factuality, while requesting a diverging suite of proposed detection techniques that should even cover cases that the model is not given at all. As a trade-off, implementations on the SHROOM dataset require the ground-truth output, since the given input does not contain the necessary semantic information to drive decisions on whether a sample is a hallucination or not. Our proposed approach only considers given *inputs* and *outputs* and does *not* probe any model, contrary to other black-box techniques (Manakul et al., 2023).

3 Task and Dataset description

Driven by upcoming challenges in the NLG landscape, SHROOM dataset focuses on the prevalent issues of models generating linguistically fluent but inaccurate (incorrect or unsupported) outputs. Participants are tasked with binary classification to identify instances of fluent overgeneration hallucinations in *model-aware* and *model-agnostic* tracks. The task encompasses three NLG domains—definition modeling (DM), machine translation (MT), and paraphrase generation (PG)—with provided checkpoints, inputs, references, and outputs for binary classification. The development set includes annotations from multiple annotators, establishing a majority vote gold label.

Data details In all cases, data follow a specific format: *src* is the input given to a model, *hyp* is the output generated by the model, *tgt* comprises the ground truth output for this specific model, *ref* indicates whether target, source or both of these fields contain the semantic information necessary to establish whether a datapoint is a hallucination, *task* refers to the task being solved and *model* to the model being used (in the model-agnostic case the *model* entry remains empty). An example of the data format is given in Table 7. Initially, 80 labeled trial samples were released, followed by unlabelled training data which contain 30k model-agnostic and 30k model-aware instances. Finally, the labeled validation set contains 499 and 501 samples for model-agnostic and model-aware settings respectively, while the test set comprises 1500 model-agnostic and 1500 model-aware labeled samples. Additional information provided in the labeled splits are *labels*, which contains a list of 'Hallucination' and 'Not Hallucination' labels as provided by 5 annotators per sample, the final *label* occurring via majority voting over the aforementioned list and *p(Hallucination)*, denoting the probability of hallucination as the percentage of agreeing annotators on the 'Hallucination' label. A thorough data analysis is provided in the App. C.

Evaluation metrics proposed from the task organizers for SHROOM are accuracy, regarding the classification success in 'Hallucination'/'Not Hallucination' classes and Spearman correlation (RHO), measuring the -positive- correlation between validation and test p('Hallucination') values.

4 Methods

As the core of our system, we propose a universal and lightweight methodology that leverages well-established pre-trained classifiers for hallucination detection. We propose 3 techniques to approach it.

4.1 Fine-tune hallucination detection model

Our first technique employs fine-tuning a pre-trained classifier dedicated to hallucination detection to learn distinguishing patterns between hallucinated/non-hallucinated SHROOM instances. More specifically, we employed a pre-trained model based on microsoft/deberta-v3-base pro-

vided by Hugging Face[4], especially designed for hallucination detection. This model was initially trained on NLI data to ascertain textual entailment. Subsequently, it underwent further fine-tuning using summarization datasets enriched with factual consistency annotations. The output of our employed model is a probability score in the [0, 1] range; a score of 0 indicates the presence of hallucination in the generated content, while a score of 1 signifies factual consistency. This probabilistic nature enables the evaluation of the model's confidence in the veracity of the generated hypotheses.

To tailor the model to the specific demands of our task, we used the provided annotated validation set of 1000 samples for training purposes. This adaptation process aimed to enhance the model's performance by aligning it with the variation and complexity present in SHROOM. Moreover, we applied a thresholding approach to make practical decisions based on the probabilistic outputs of the model. By setting a threshold at 0.5, we categorize predictions with scores above this threshold as indicative of input-output consistency, while the rest are considered as potential hallucinatory instances.

4.2 Fine-tune NLI models

In the context of detecting hallucinated answers, we also employed NLI models, an approach that has witnessed significant advancements, while investigating semantic intricacies close to hallucinations. NLI models play a crucial role in enabling comprehension of the sophisticated connections between sentences, categorizing the relationship between a *hypothesis* and a *premise* into entailment, neutral, or contradiction. In terms of our task, we convert hallucination detection to an NLI problem: given the input (termed as *hypothesis-hyp*) to a model and the premise (named *target-tgt*) we evaluate whether *tgt* entails, contradicts or remains neutral to *hyp*.

To execute this approach in technical terms, we select a pre-trained NLI model available through Hugging Face[5]. This model, based on mDeBERTa-v3-base architecture, was originally trained on a large-scale multilingual dataset, making it well-suited for handling diverse linguistic details. To fine-tune the NLI model and tailor it to the specific intricacies of our task, we employed the annotated validation set, as in the previous case.

[4] https://huggingface.co/vectara/hallucination_evaluation_model

[5] https://huggingface.co/MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7

4.3 Voting Classifier

In our final approach, we employed an ensemble technique known as a Voting Classifier. The underlying principle is to aggregate the collective insights derived from each constituent classifier (in our case the previously mentioned models), ultimately predicting the output class based on the highest majority of votes. By doing so, the ensemble not only leverages the individual strengths of each method but also mitigates potential weaknesses, thereby enhancing the overall predictive performance in a deliberate effort to address the inherent complexity and variability within the dataset, contributing to a more nuanced and accurate understanding of the phenomena under investigation.

5 Experiments

5.1 Experimental setup

All our experiments were executed using Google Colab platform with a single Tesla T4 GPU.

Fine-tune hallucination model Our fine-tuned model underwent a rigorous training and evaluation process, utilizing SHROOM data provided by the task organizers. Specifically, the model was trained with the annotated validation set and evaluated against the trial set. In the pre-processing phase, from each data point, we extracted the *hyp* and *tgt* components to serve as inputs to the model.

To optimize the model's performance in terms of both accuracy and p('Hallucination'), we implemented a dual-training strategy. The model was trained twice, employing binary labels (0 for Hallucination and 1 for Not Hallucination) in one iteration and float labels (representing 1-p('Hallucination')) in the other. This dual-training approach allowed us to derive two crucial aspects from the model: the binary label indicating the presence or absence of hallucination, and the corresponding probability score indicating the likelihood of hallucination. The hyperparameters for fine-tuning are comprehensively detailed in Table 1.

Hyperparameter	Value
train dataloader	validation set (1,000 samples)
evaluator	trial set (80 samples)
epochs	5
evaluation steps	10,000
warm-up steps	10% of train data for warm-up

Table 1: Hyperparameters used for the hallucination detection model fine-tuning

Natural Language Inference (NLI) models This NLI model was already trained with the multilingual-nli-26lang-2mil7 (Laurer et al., 2022) dataset and the XNLI validation dataset (Conneau et al., 2018), both containing three different labels: 'entailment', 'neutral' and 'contradiction'. During the training phase, we systematically mapped the 'Hallucination' label to 'contradiction' and the 'Not Hallucination' label to 'entailment', ensuring a binary representation of the hallucinatory nature of the content. This transformation facilitated the training process by providing clear labels for the model to learn the distinctions between hallucinatory and non-hallucinatory instances.

Post-training, the model's predictions were assessed using the entailment score, and a strategically chosen threshold was employed to distinguish between hallucinations and non-hallucinations. Prior to training, we experimented with a wide range of threshold values, concluding that a threshold of 0.8 optimized the accuracy of the trial set. Simultaneously, for the determination of the percentage of Hallucination for each data point, we used the entailment percentage subtracted from 1.

A detailed account of the parameters employed for training this NLI model is outlined in Table 2.

Hyperparameter	Value
train dataset	validation set (1,000 samples)
learning rate	2e-05
epochs	5
warm-up ratio	0.06
weight decay	0.01

Table 2: Hyperparameters used for NLI fine-tuning

Voting Classifier In the final leg of our methodological exploration, the Voting Classifier integrates the pre-trained hallucination detection model, its fine-tuned counterpart from §4.1, and the fine-tuned NLI model described in §4.2.

The Voting Classifier operates on a dual strategy for hallucination categorization. First, for the binary labels, we assigned the majority label ('Hallucination' or 'Not Hallucination') among the three models to each data point. Second, to determine the percentage of hallucination for each data point, we provided two methodologies. For the first one, we implemented a similar methodology to the one used in the validation and trial sets, i.e. the percentage of hallucination derived from the majority vote of the annotators. By emulating the same process, we calculate the percentage of models that labeled a given data point as 'Hallucination'. For the second one, we use the float p('Hallucination') scores of each of the three models constituting the ensemble and extract the average value.

5.2 Results

Baseline System During the evaluation phase, we were provided with a baseline system, which was based on a simple prompt retrieval approach, derived from SelfCheck-GPT(Manakul et al., 2023), using an open-source Mistral instruction-tuned model as its core component (the prompt is shown in Table 6). If the answer starts with'Yes' the sample is classified as 'Not Hallucination' with p('Hallucination') equal to the probability that the token was chosen subtracted from 1, else if the answer starts with 'No' the sample is classified as 'Hallucination' with p('Hallucination') equal to the probability that the token was chosen. If the answer starts with neither, the label is assigned randomly and p('Hallucination') equals to 0.5.

Averaged results for all our experiments are presented in Table 3. The Voting Classifier achieves top results, with a more notable difference in the model-agnostic setting. This is an expected behavior since the ensembling of models is designed to boost the performance of its standalone constituents.

Method	acc.↑	rho↑
Model-aware		
Baseline Model	0.745	0.488
Fine-tune hal-detect model	0.795	0.685
NLI model	0.77	0.591
Voting Classifier-majority vote	**0.799**	0.691
Voting Classifier-averaged percentage	**0.799**	**0.693**
Model-agnostic		
Baseline Model	0.697	0.402
Fine-tune hal-detect model	0.778	0.668
NLI model	0.751	0.548
Voting Classifier-majority vote	**0.78**	0.632
Voting Classifier-averaged percentage	**0.78**	**0.643**

Table 3: Final results for model-aware and model-agnostic variants. **Bold** denotes best results. The two Voting Classifiers differentiate from the method applied to calculate the p('Hallucination) as explained in 5.1

We demonstrate the computational efficiency of our proposed methods regarding the training and inference time needed in Table 4. The Voting Classifier sums the times of all three of its model-voters. Since reported runtimes were achieved using the T4 GPU of the free Google Colab version, our proposed methods can be replicated and utilized by

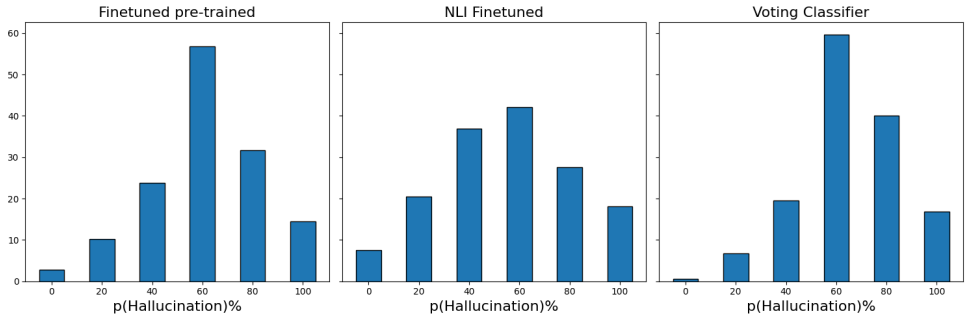


Figure 1: p('Hallucination') for all misclassified samples of model aware dataset.

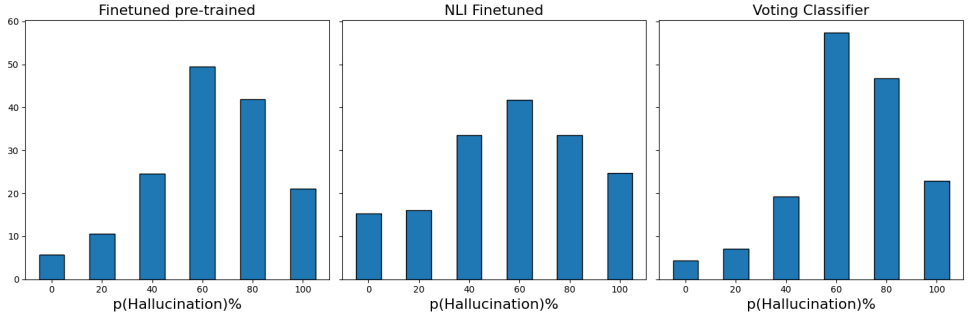


Figure 2: p('Hallucination') for all misclassified samples of model agnostic dataset.

any user, without any budget or time limitations, nor the need to access sophisticated hardware.

Method	**Training↓**	**Inference↓**
pre-trained hal-detect model	-	39.00
Fine-tune hal-detect model	91.59	45.66
NLI model	927.14	58.96
Voting Classifier	1,018.73	143.62

Table 4: Training and inference time in seconds.

NLG Model	**Task**	**aware-acc↑**
Hal-detect model fine-tuning		
tuner007/pegasus_paraphrase	PG	0.856
facebook/nllb-200-distilled-600M	MT	0.824
ltg/flan-t5-definition-en-base	DM	0.724
NLI model fine-tuning		
tuner007/pegasus_paraphrase	PG	0.803
facebook/nllb-200-distilled-600M	MT	0.789
ltg/flan-t5-definition-en-base	DM	0.703
Voting Classifier		
tuner007/pegasus_paraphrase	PG	0.861
facebook/nllb-200-distilled-600M	MT	0.828
ltg/flan-t5-definition-en-base	DM	0.73

Table 5: Model-aware accuracy per model and task.

Moreover, per-task and model hallucination detection for the model-aware dataset is presented in Table 5. The PG task demonstrates superior performance compared to the other two tasks, while the DM task reports significantly lower accuracy. This disparity in outcomes can be explained by the inherent characteristics of each task when formulated as a paraphrase problem. The PG task exhibits notably higher results owing to its direct alignment with the paraphrase objective. Similarly, the MT task, which evaluates translations from the LLM against ground truth translation, achieves relatively comparable results. Conversely, the DM task faces the complexities of articulating precise and contextually relevant definitions. Consequently, the DM task exhibits notably lower accuracy due to the intricacies of handling more complex sentence structures. The Voting Classifier remains the top scorer in each of the tasks, highlighting the power of ensembling individual predictors.

Finally, we perform *some error* analysis on the misclassified samples (Figures 1, 2): we measure the p('Hallucination') for misclassifications for all our 3 methods. Ideally, p('Hallucination') values for misclassifications should lie close to the discrimination threshold of 0.5, indicating that their separability is highly uncertain. Indeed, our best performing Voting Classifier presents a peak for p('Hallucination')=0.6 for both model-aware and model-agnostic settings, highlighting that misclassified samples are in any case hard to classify in their correct class. Moreover, the p('Hallucination') values in the range [0.0-0.4] - corresponding to the 'Not Hallucination' label- are lower for the Voting Classifier in comparison to the other two models, denoting that ensembling

reduces misclassifications for non-hallucinatory instances.

6 Conclusion

In this work, we detect and analyze hallucinations from the SHROOM dataset introduced in SemEval 2024 Task 6. We propose a computationally efficient methodology based on fine-tuning models that present semantic cues close to SHROOM's hallucinations, while model ensembling further boosts results in 3 NLG tasks. Our techniques operate in a fully black-box setting, solely requiring inputs and outputs obtained from NLG models. Our error analysis demonstrates that our misclassifications are samples of high uncertainty in terms of hallucination probability and, therefore hard to be discerned overall. In total, we aspire that our simple though efficient technique will assist future research in the crucial hallucination detection field.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying.

Zouying Cao, Yifei Yang, and Hai Zhao. 2023. Autohall: Automated hallucination dataset generation for large language models. *ArXiv*, abs/2310.00259.

Xiang Chen, Duanzheng Song, Honghao Gui, Chenxi Wang, Ningyu Zhang, Jiang Yong, Fei Huang, Chengfei Lv, Dan Zhang, and Huajun Chen. 2023. Factchd: Benchmarking fact-conflicting hallucination detection. *ArXiv*, abs/2310.12086.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai – a tool augmented framework for multi-task and multi-domain scenarios.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Hanyu Duan, Yi Yang, and Kar Yan Tam. 2024. Do llms know about hallucination? an empirical investigation of llm's hidden states.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Kasper Welbers. 2022. Less Annotating, More Classifying – Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT - NLI. *Preprint*. Publisher: Open Science Framework.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Factuality enhanced language models for open-ended text generation.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*,

Mexico City, Mexico. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2024. Generating benchmarks for factuality evaluation of language models.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models.

Ziwei Xu, Sanjay Jain, and Mohan S. Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *ArXiv*, abs/2401.11817.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *ArXiv*, abs/2309.01219.

A Organizers' baseline

The prompt used by the organizers to construct the baseline Mistral instruction-tuned model is demonstrated in Table 6.

Prompt
Context {tgt} Sentence: {hyp} Is the sentence supported by the context above? Answer Yes or No:

Table 6: Prompt used in the Baselined System

B Data format

In Table 7 we present some examples from the unlabelled training dataset containing model-agnostic and model-aware instances. Regarding the machine translation (MT) task, we could detect a variety of languages, including Russian, Arabic, Chinese, Yorùbá, Telugu, Tsonga, Uzbek, Sinhalese, Quechuan, Mizo and others. Language information was not provided, so we manually explored the *src* samples in terms of linguistic variability.

Model-agnostic definition modeling (DM) hypotheses contain some 'qualifiers', which may guide a model under usage to return a more suitable definition. For example, in the context of the hypothesis containing the definition "(obsolete) An odour," the term "obsolete" indicates that the provided definition is no longer in common use or is outdated. The word "obsolete" is used as a qualifier to convey that the term or concept being defined, in this case, "An odour," was once used to represent a specific meaning but is no longer considered current or applicable in contemporary language.

Another notable observation is that model-aware paraphrase-generation (PG) does not contain any information in *tgt*.

C Exploratory data analysis

Trial set We explore the frequency of each task occurring within samples from different dataset splits, commencing from the initially released trial set. In Figure 3 we present the task distribution of the first 80 trial samples.

Unlabelled data (training set) Figure 4 represents the distribution in the training set. In both model-agnostic and model-aware settings each task contains an equal number of samples (10k samples per task in each setting). In our methodologies, we abstained from utilizing the provided unlabeled training dataset as it did not align with our main approaches.

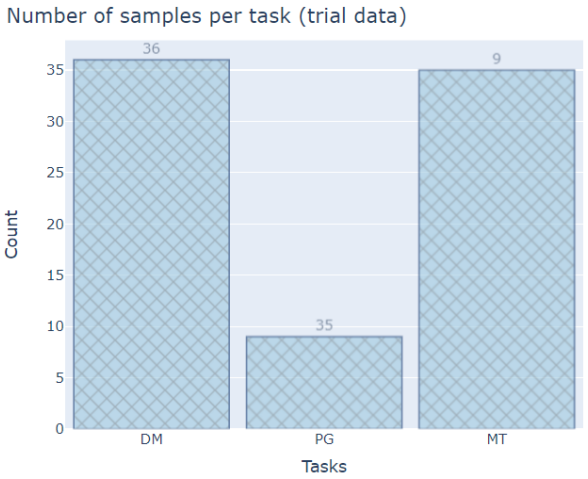
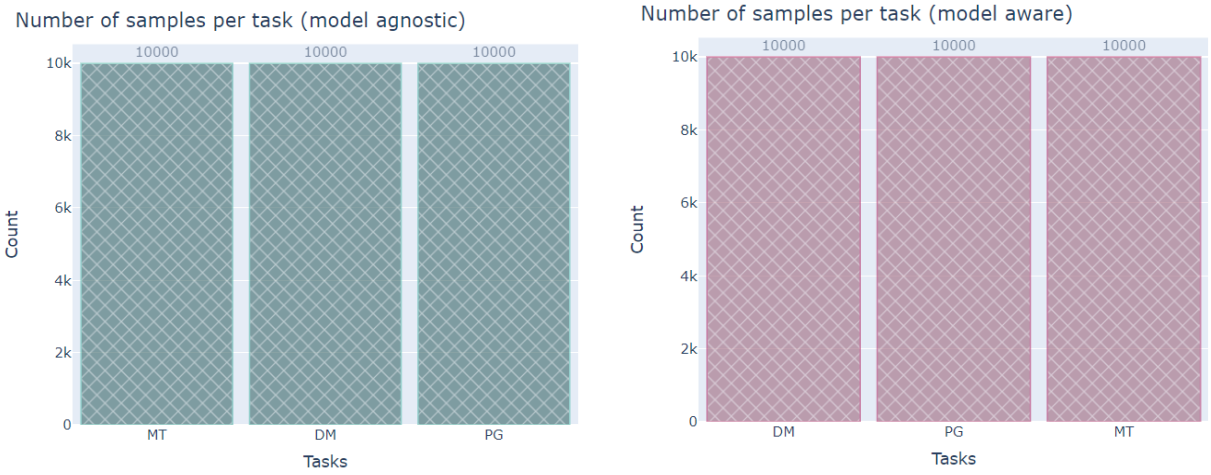


Figure 3: Distribution of per task samples in the initially released trial set.

Validation set Moving on to labeled data, we commence with the validation (dev) set, for which we present per task distributions in Figure 5. We observe a difference in the distribution of labels

Model-agnostic	
Machine Translation	'hyp': "Don't worry, it's only temporary.", 'tgt': "Don't worry. It's only temporary.", 'src': 'Не волнуйся. Это только временно.', 'ref': 'either', 'task': 'MT', 'model': ''
Definition modelling	'hyp': '(uncountable) The quality of being oronymy; the state of being oronymy.', 'tgt': 'The nomenclature of mountains, hills and other geographic rises.', 'src': 'An ancient survival in Turkish <define> oronymy </define> is quite possible , but I have not found Nihan Dag on the relevant sheets of the 1 : 200,000 map of Turkey , which are very detailed in matters of oronymy ;', 'ref': 'tgt', 'task': 'DM', 'model': ''
Definition modelling	'hyp': '(intransitive, obsolete) To make a magazin of; to compose a magazin.', 'tgt': '(colloquial) The act of editing or writing for a magazine.', 'src': "Thus , though Byron is gone after his Don Juan — Scott and Southey out of the rhyme department — Wordsworth stamp - mastering — Coleridge 's poetry in abeyance — Crabbe mute as a fish - Campbell and Wilsont merely <define> magazining </define>", 'ref': 'tgt', 'task': 'DM', 'model': ''
Paraphrase Generation	'hyp': 'You got something for me, huh?', 'tgt': '', 'src': 'Got something for me?', 'ref': 'src', 'task': 'PG', 'model': ''
Model-aware	
Machine Translation	'hyp': "It's like pushing a heavy wheel up a mountain. It splits the nucleus again and releases some energy.", 'tgt': 'Sort of like rolling a heavy cart up a hill. Splitting the nucleus up again then releases some of that energy.', 'src': '有像把沉重的手推推上山。再次分裂核子然後放一些能量', 'ref': 'either', 'task': 'MT', 'model': 'facebook/nllb-200-distilled-600M'
Machine Translation	'hyp': 'Our Mailoamiris of the System of Treatment of Ulilaes have created a place for these little ones.', 'tgt': 'We perceive the Foster Care System to be a safety zone for these children.', 'src': 'Maamiris tayo a ti Sistema iti Panangtaripato kadagiti Ulila ket natalged a lugar para kadagitoy nga ubbing.', 'ref': 'either', 'task': 'MT', 'model': 'facebook/nllb-200-distilled-600M'
Definition modeling	'hyp': 'To be obsequiously interested in .', 'tgt': '(usually followed by over or after) To fuss over something adoringly ; to be infatuated with someone .', 'src': "Sarah mooned over sam 's photograph for months . What is the meaning of moon ?", 'ref': 'tgt', 'task': 'DM', 'model': 'ltg/flan-t5-definition-en-base'
Paraphrase Generation	'hyp': "Mr Barros Moura's report looks to the future in my opinion.", 'tgt': '', 'src': 'In my opinion, the most important element of the report by Mr Barros Moura is that it looks to the future.', 'ref': 'src', 'task': 'PG', 'model': 'tuner007/pegasus_paraphrase'

Table 7: Examples from the unlabelled training set.



(a) Model-agnostic sample distribution in the training set.

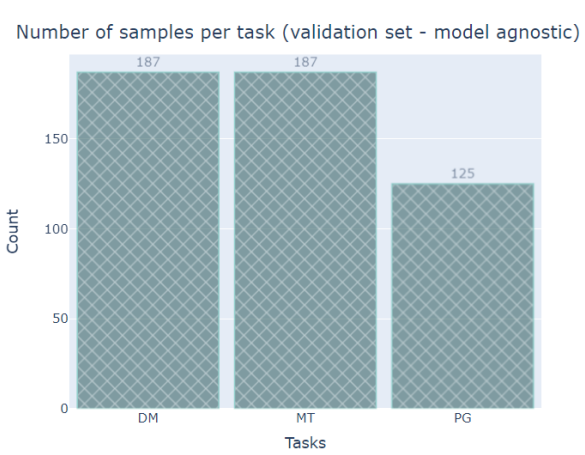
(b) Model-aware sample distribution in the training set.

Figure 4: Distribution of unlabelled training samples per task in both model-agnostic and model-aware settings.

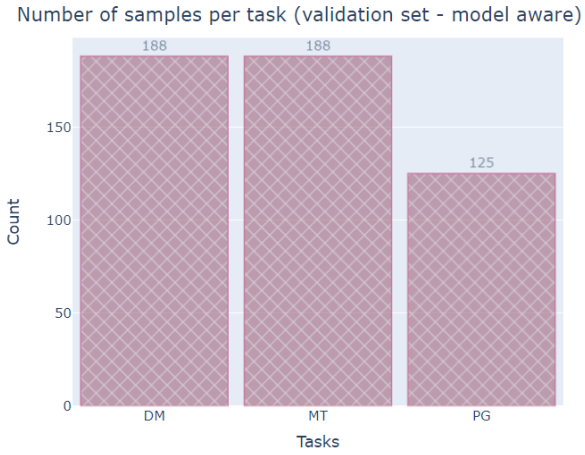
in comparison to the balanced training set distribution of Figure 4; nevertheless, since we do not exploit any unlabelled instance, this does not pose a limitation for us at this point.

We proceed with studying the validation set label distribution. Related results are presented in Figure 6, denoting label imbalance in both model-agnostic and model-aware settings.

The distribution of hallucination probability is presented in Figure 7. As expected, low p('Hallucination') values are more common (indicating that fewer annotations voted for the presence of a hallucinatory instance), since 'Not Hallucination' is the majority label in both settings. Ideally, we wish borderline probabilities to be low: The highest the disagreement for a certain sample, the

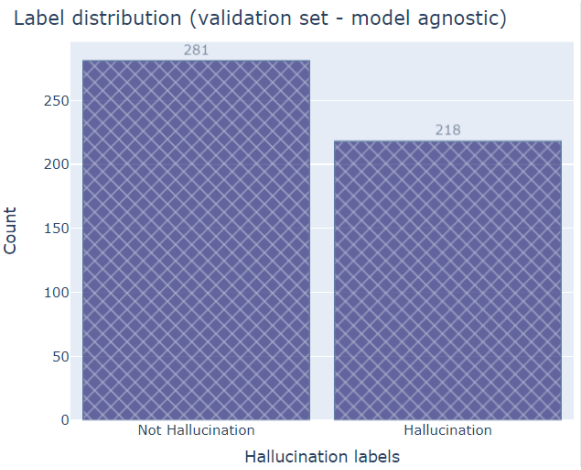


(a) Model-agnostic sample distribution in the validation set.

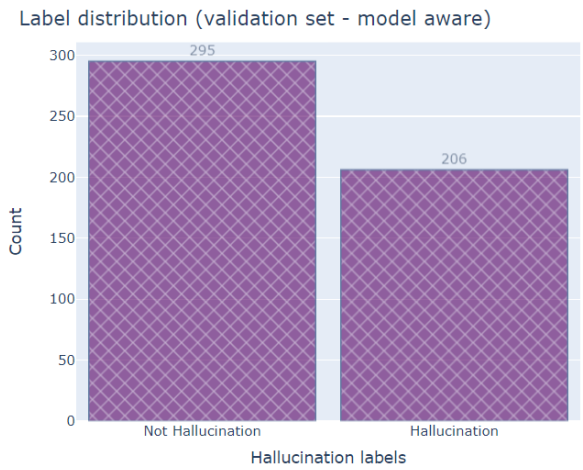


(b) Model-aware sample distribution in the validation set.

Figure 5: Distribution of labeled validation samples per task in both model-agnostic and model-aware settings.

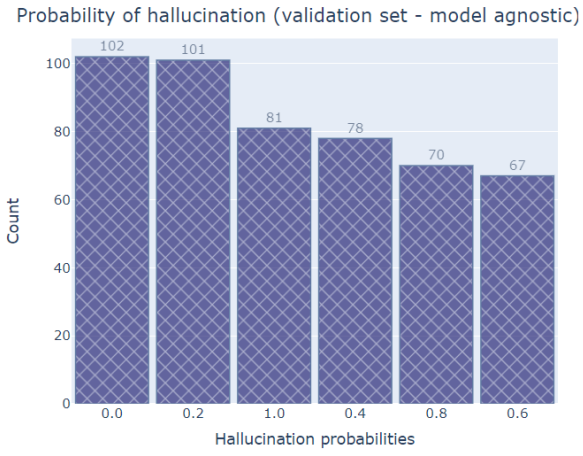


(a) Model-agnostic label distribution in the validation set.

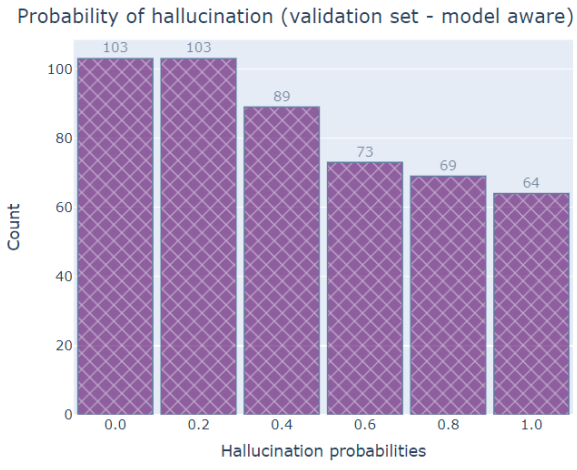


(b) Model-aware label distribution in the validation set.

Figure 6: Distribution of validation labels in both model-agnostic and model-aware settings.

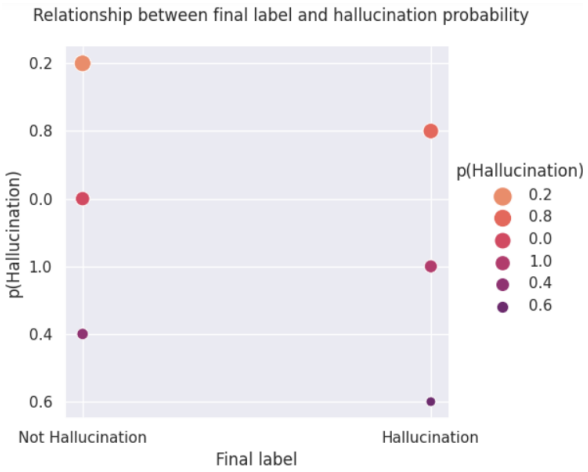


(a) Model-agnostic hallucination probability distribution in the validation set.

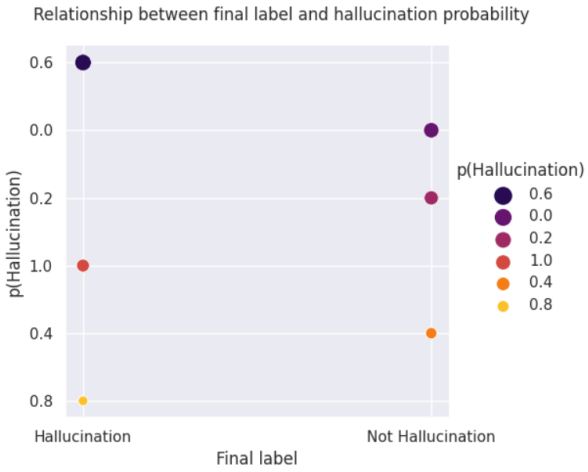


(b) Model-aware hallucination probability distribution in the validation set.

Figure 7: Distribution of hallucination probability (majority voting among human annotators' labeling) in both model-agnostic and model-aware settings in the validation set.

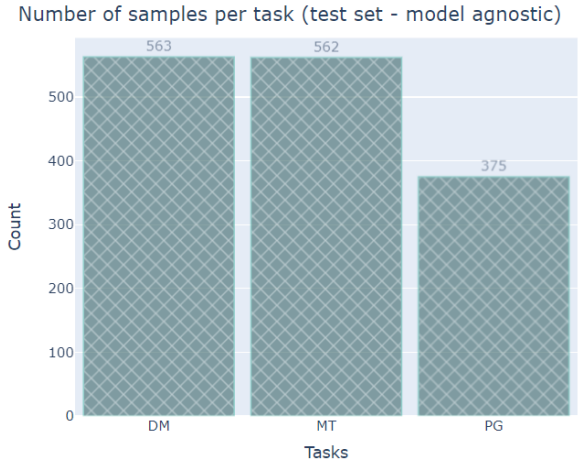


(a) Hallucination probability per label (Model-agnostic).

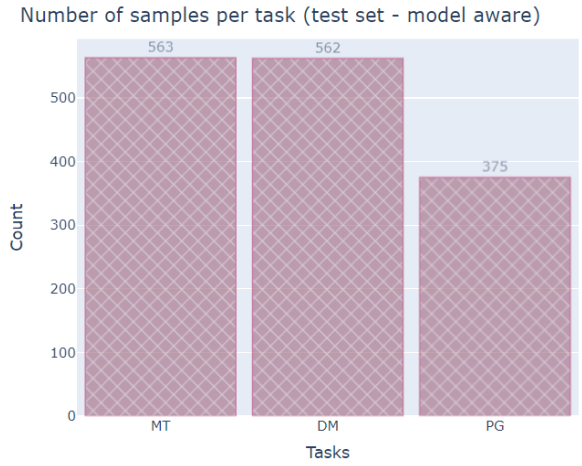


(b) Hallucination probability per label (Model aware).

Figure 8: Distribution of hallucination probability in each validation label ('Hallucination' vs 'Not Hallucination'). Annotators significantly agree on whether a sample contains a hallucination or not.

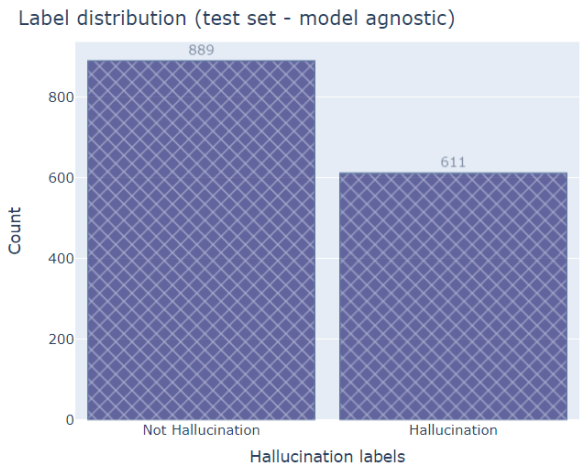


(a) Model-agnostic sample distribution in the test set.

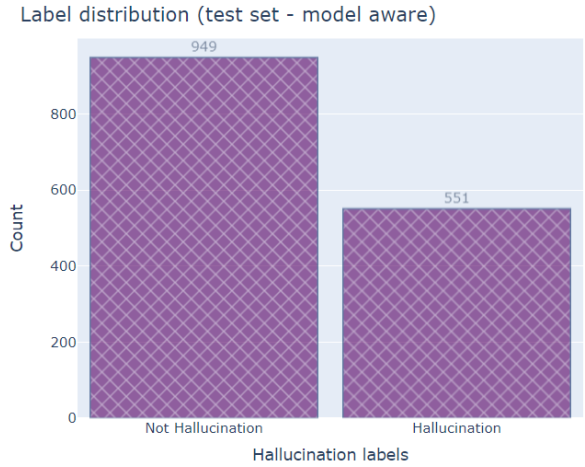


(b) Model-aware sample distribution in the test set.

Figure 9: Distribution of labeled test samples per task in both model-agnostic and model-aware settings.

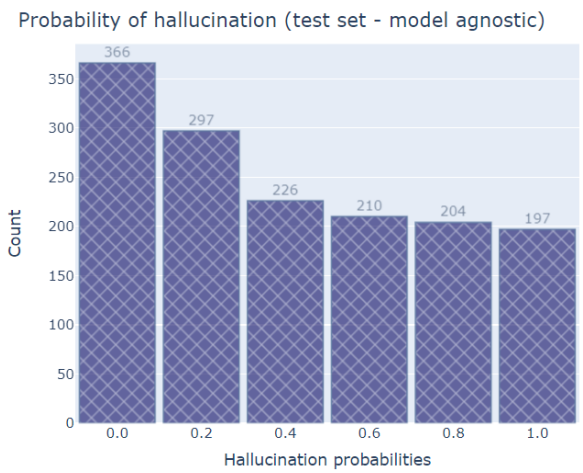


(a) Model-agnostic label distribution in the test set.

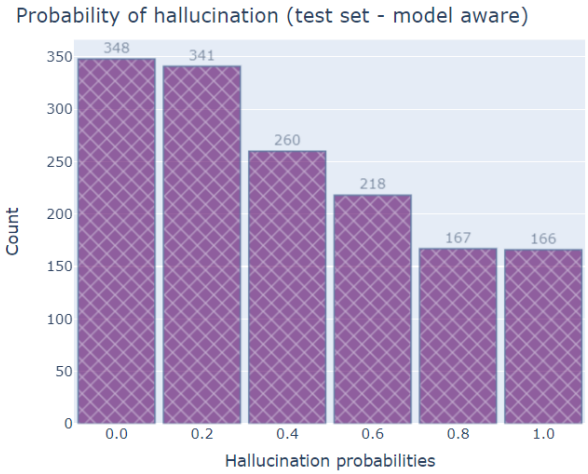


(b) Model-aware label distribution in the test set.

Figure 10: Distribution of test labels in both model-agnostic and model-aware settings.

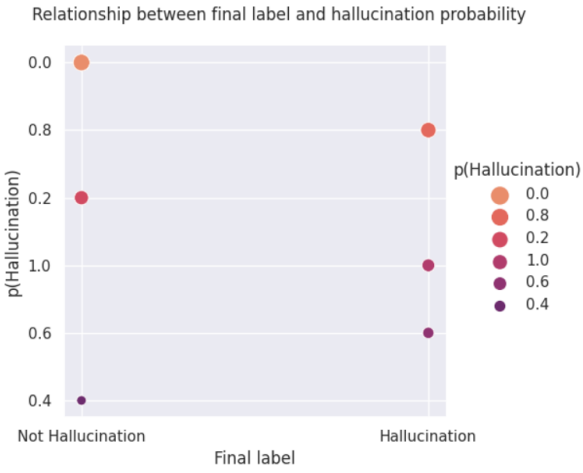


(a) Model-agnostic hallucination probability distribution in the test set.

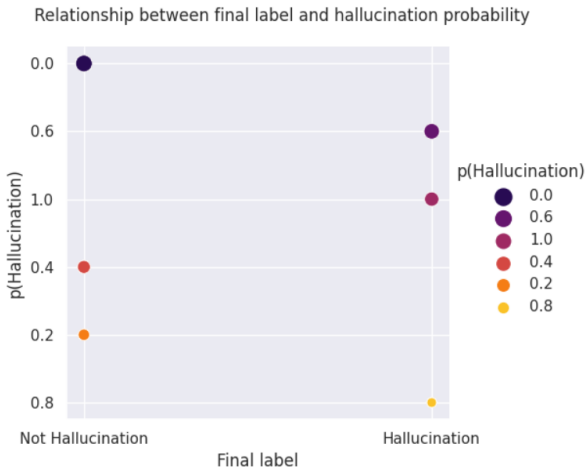


(b) Model-aware hallucination probability distribution in the test set.

Figure 11: Distribution of hallucination probability (majority voting among human annotators' labeling) in both model-agnostic and model-aware settings in the test set.



(a) Hallucination probability per label (Model-agnostic).



(b) Hallucination probability per label (Model aware).

Figure 12: Distribution of hallucination probability in each test label ('Hallucination' vs 'Not Hallucination). Annotators significantly agree on whether a sample contains a hallucination or not.

closest to the 0.5 threshold the hallucination probability will be (a p('Hallucination')=0.4 denotes that 3/5 annotators voted for 'Not Hallucination', while the rest 2/5 voted for the opposite; on the other hand, a p('Hallucination')=0.6 denotes that 3/5 annotators voted for 'Hallucination', while the rest 2/5 voted for 'Not Hallucination'. Therefore, the highest uncertainty is observed close to the 0.5 boundary). This requirement is adequately satisfied especially in the model-agnostic case (left plot of Figure 7), where p('Hallucination')=0.6 is the least frequent.

Further insights can be obtained by looking at Figure 8: when smaller dots are assigned to probabilities close to the 0.5 threshold, the annotators' disagreement is lower, therefore classifying a sample as 'Hallucination' or 'Not hallucination' is less uncertain. Indeed, the less frequently appearing p('Hallucination')=0.4 and p('Hallucination')=0.6 values in the model-agnostic case denote high separability between hallucinated and non-hallucinated samples. However, highly certain values, such as p('Hallucination')=0.0 and p('Hallucination')=1.0 only rank in the middle, therefore even if samples are separable with low uncertainty, some minor disagreement persists (1/5 annotators frequently disagrees with the rest). Overall, annotators are almost equally confident in classifying 'Hallucination' and 'Not Hallucination' samples, as indicated by the matching pattern regarding label uncertainty

for both labels. The model-aware case is more confusing, with p('Hallucination')=0.6 scoring the highest; therefore, classifying a sample as 'Hallucination' is often accompanied by high uncertainty. On the contrary, uncertainty is lower for the 'Not Hallucination' label, with p('Hallucination')=0.0 ranking as the second most frequent probability. We can conclude that in the model-aware setting of the validation set, annotators are more confident in recognizing the 'Not Hallucination' class in comparison to the 'Hallucination' one.

Test set As for the test set, Figure 9 represents the number of samples per task for both settings. Note that the test task distribution is similar to the validation distribution of Figure 5with PG being a minority label in all cases.

In terms of ground-truth label (Hallucination vs Not Hallucination), Figure 10 highlights some label imbalance, rendering the prediction of 'Not Hallucination' more possible in a random setup for both model-agnostic and model-aware settings. This label distribution matches the validation set label distribution (Figure 6), for which 'Not Hallucination' was the majority class as well.

Hallucination probability per setting is depicted in Figure 11, with lower hallucination values in the range [0, 0.2) being more common. This is again somehow expected since 'Not Hallucination' is the majority class in test labels. More insights can be obtained by looking at Figure 12, which relates the hallucination probability with the label. Especially in the model-agnostic setting (Figure 12 - left), the p('Hallucination')=0.4 and p('Hallucination')=0.6 values are the lowest (smaller dots), while p('Hallucination')=0.0 is the highest, denoting that annotators are often certain regarding non-hallucinated samples. Certainty for hallucinated samples is somehow lower, as p('Hallucination')=1.0 lies somewhere in the middle. Nevertheless, p('Hallucination')=0.8 is the second more frequent value denoting that 4/5 annotators frequently annotate a sample as 'Hallucination'. By observing the right plot of Figure 12, we conclude that certainty is lower in the model-aware setting. Even though p('Hallucination')=0.0 remains the most frequent probability, indicating high agreement regarding non-hallucinated samples, the p('Hallucination')=0.6 value stands in the second place. Therefore, many samples classified as 'Hallucination' achieved this label with low agreement (3/5 annotators). Also, the p('Hallucination')=0.2 and p('Hallucination')=0.8 are the lowest, denoting that higher agreement (4/5 annotators agreeing) is rare for both 'Hallucination' and 'Not Hallucination' labels. We can assume that model-aware samples are harder by nature to be classified in any of the labels.

D NLI-Hyperparameters

The hyperparameters utilized for the NLI model fine-tuning mirrored those employed during the training of the initial model. The selection of hyperparameters followed a series of experiments, which yielded significantly lower levels of accuracy. Some of the experiments are displayed in the Table 8

epochs	lr	warmup ratio	weight decay	accuracy
5	2e-05	0.06	0.01	0.83
10	2e-06	0.1	0.01	0.75
5	2e-04	0.01	0.05	0.53
5	2e-05	0.05	0.001	0.8
5	2e-06	0.08	0.1	0.79

Table 8: Accuracy on trial-set from experiments with hyperparameters. The first row displays the hyperparameters chosen for finetuning

JMI at SemEval 2024 Task 3: Two-step approach for multimodal ECAC using in-context learning with GPT and instruction-tuned Llama models

Arefa[1,†], **Mohammed Abbas Ansari**[1,†], **Chandni Saxena**[2], **Tanvir Ahmad**[1]
[1]Jamia Millia Islamia University, New Delhi, India
[2]The Chinese University of Hong Kong, Hong Kong SAR, China
{arefa2001, mohd.abbas.ansari.2001}@gmail.com
csaxena@cse.cuhk.edu.hk, tahmad2@jmi.ac.in

Abstract

This paper presents our system development for SemEval-2024 Task 3: "The Competition of Multimodal Emotion Cause Analysis in Conversations". Effectively capturing emotions in human conversations requires integrating multiple modalities such as text, audio, and video. However, the complexities of these diverse modalities pose challenges for developing an efficient multimodal emotion cause analysis (ECA) system. Our proposed approach addresses these challenges by a two-step framework. We adopt two different approaches in our implementation. In Approach 1, we employ instruction-tuning with two separate Llama 2 models for emotion and cause prediction. In Approach 2, we use GPT-4V for conversation-level video description and employ in-context learning with annotated conversation using GPT 3.5. Our system wins rank 4, and system ablation experiments demonstrate that our proposed solutions achieve significant performance gains. All the experimental codes are available on Github.

1 Introduction

Emotion Cause Analysis (ECA) is centered around the extraction of potential cause clauses or pairs of emotion clauses and cause clauses from human communication, enabling a deeper understanding of communication dynamics. By incorporating multimodal cues like visual scenes, facial expressions, and vocal intonation, it facilitates a comprehensive and technically robust analysis of the factors that trigger diverse emotional reactions (Mittal et al., 2021; Zhang and Li, 2023; Zheng et al., 2023b). Despite the considerable amount of research conducted using diverse audio, visual, and text modalities (Gui et al., 2018; Xia and Ding, 2019; Fan et al., 2020; Shoumy et al., 2020; Abdullah et al., 2021), there has been a noticeable gap in the exploration of multimodal ECA in natural settings (human conversations). In this context, Wang et al. (2023a) introduce Multimodal Emotion Cause Analysis in Conversations (ECAC) task and provide Emotion-Cause-in-Friends (ECF) dataset, which incorporates text, audio, and video modalities. This task consists of two sub-tasks: Textual Emotion-Cause Pair Extraction in Conversations (Subtask 1) and Multimodal Emotion Cause Analysis in Conversations (Subtask 2). A detailed description of these sub-tasks can be found in the task description paper (Wang et al., 2024a).

In our submission to Subtask 2 of multimodal ECAC, this paper presents two distinct approaches to address the ECAC problem, giving competitive results. Drawing inspiration from the effectiveness of LLMs in diverse downstream tasks (Wang et al., 2023b, 2024b; Yang et al., 2024), including emotion recognition, we propose two LLM-based approaches that decompose the emotion-cause pair extraction process into two steps. The first step involves predicting the emotions of the utterances in the conversation. In the next step, we utilize these emotion labels to guide cause extraction. **Approach 1** involves instruction-tuning two separate Llama 2 models for emotion and cause prediction, while **Approach 2** leverages the in-context learning (ICL) capabilities (Dong et al., 2023) of the GPT-3.5 model. Additionally, we introduce an efficient technique using the GPT-4V model to extract conversation-level descriptions from video modality.

During the evaluation, our team ranked 4th on the leaderboard competing against more than 25 teams with a weighted-F1 score of 0.2816.

2 Background

2.1 Task definition

The input for the task, D, comprises N conversations. As described by Wang et al. (2023a), given a

[†]Equal contribution

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1561–1576

conversation $D_i = \{u_1, u_2, \ldots, u_M\}$ consisting of M utterances, where each utterance is represented by text, audio, and video, i.e. $u_j = [t_j, a_j, v_j]$, the goal of the task is to extract a set of emotion-cause pairs $P = \{\ldots, (u_k^e, u_k^c), \ldots\}$, where u_k^e denotes an emotion utterance and u_k^c corresponds to the cause utterance.

2.2 Related Work

The detailed Related Work section can be found in Appendix A.

2.3 Dataset

We use the Emotion-Cause-in-Friends (ECF) dataset provided by Wang et al. (2023a), which is summarized in Table 1. This dataset contains 13,509 multimodal utterances that occur in the American sitcom *Friends* with 9272 emotion-cause pairs. Each utterance consists of the text, video, and audio.

Class-distribution The dataset is imbalanced as shown in Fig. 1 wherein around 44% of the utterances have neutral emotion. Disgust and Fear constitute only 3% and 2.7% of the emotions.

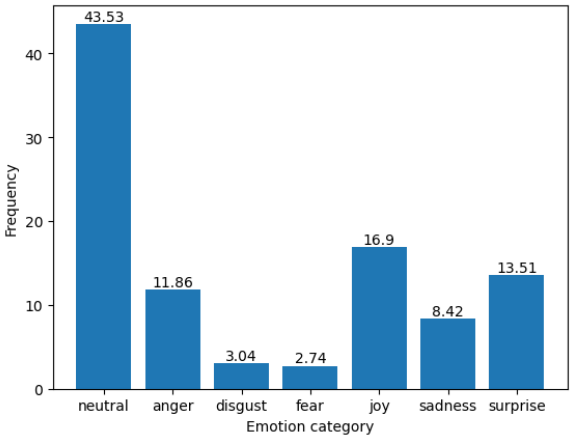


Figure 1: Percentage of each of the seven emotion categories

Relative positions of emotion and causes Interestingly, 49.95% of the causes are self-causes meaning that the same utterance caused itself as shown in Fig. 2. This is also intuitive, as what one speaks or expresses often elicits the emotion of their utterance. Note that the dataset curators have also annotated utterances coming after the emotion utterance as its cause. These constitute only about 2.8% of all causes and are one or two utterances away. 94.95% of the causes are 0-5 utterances behind the emotion utterance. The fact that what you speak or other interlocutors in the conversation speak affects the emotion of subsequent utterances explains this phenomenon.

Items	Number
Conversations	1344
Utterances	13,509
Emotional Utterances	7,690
Self-Causal Utterances	4,892
Non-Self-Causal Utterances	2,189
No Cause Emotional Utterances	609
Later-Causal Utterances	177

Table 1: Statistics of causes for emotional utterances.

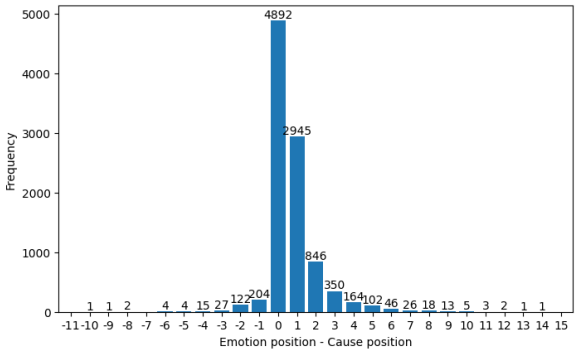


Figure 2: Relative position of emotion and causes

3 Methodology

3.1 Overview

We treat the task at hand as a two-step process. In the first step, we predict the emotion of each utterance in all N conversations. Here, the context C_j for utterance u_j of conversation D_i is the entire conversation itself. Given E target emotion labels and $\hat{y_j^e}$ as the predicted emotion label, the problem can be formulated as (where θ denotes the parameters):

$$\hat{y_j^e} = \arg\max_e \mathcal{P}(y^e | u_j, C_j, \theta) \quad (1)$$

In the second step, given these emotion labels, we predict the causes of each utterance that has an emotion other than neutral. The causes will be a subset of all utterances in the conversation D_i. Let the learned function be $f : U \rightarrow 2^U$, where U is the set of all utterances in the given conversation. It predicts the subset $\hat{y_j^c}$ of cause of emotion utterance u_j where $\hat{y_j^e} \neq$ neutral as:

$$\hat{y_j^c} = \arg\max_{y^c \in 2^U} \mathcal{P}(y^c | u_j, \hat{y_j^e}, C_j, \theta) \quad (2)$$

3.2 Approach 1: Fine-tuned Llama-2

In our first approach, we perform instruction fine-tuning of the Llama 2 Large Language Model, an open-source model developed by GenAI, Meta (Touvron et al., 2023). From the three variants with 7, 13, and 70 billion parameters, we use the 13 billion parameter model due to resource constraints, albeit the performance of this model achieves state-of-the-art results on various downstream NLP tasks compared to other models of similar sizes (Touvron et al., 2023). In addition, we use the Llama 2-chat version of the model[1], which is optimized for dialogue use cases as it aligns with our task. In our approach, we use Llama2 API[2] for prompt engineering. Through zero-shot prompting, we select optimal prompts for emotion identification and cause prediction. We observed that treating these two tasks separately resulted in better model output. This approach involves first identifying the emotions of all utterances in the conversation. We then add these emotion labels to the conversation and prompt the model to predict the causes for each emotion utterance. Consequently, we perform supervised fine-tuning of two separate Llama 2 models for these tasks. Although this increases the inference time, the significant performance gains outweigh the introduced latency. We treat both tasks as conditional generation, where the model generated the emotion label in the first case and the cause list in the second case, given the prompt. Detailed explanations of these approaches are provided in the following sections. The fine-tuning procedure is shown in Fig.3.

3.2.1 Emotion recognition

To perform emotion recognition, we create a dataset where each sample includes an utterance u_j from one of the N conversations D for which the LLM needs to output the emotion label. We incorporate the entire conversation D_i along with speaker information as context in our prompt. This contextual information enhances the model's understanding of the flow of emotions within the conversation, as demonstrated by our ablation studies in Section 5. The instruction I_j^e, which gave the best results, is given in Appendix E.1 along with detailed prompt examples. The prompt consists of the instruction I_j^e and the context C_j for utterance u_j:

$$Prompt_j = (C_j, I_j^e) \tag{3}$$

Using this prompt as the input and the corresponding true emotion label y_j^e, we perform supervised fine-tuning of a Llama 2-13b model.

$$\hat{y_j^e} = \textbf{Llama}_e(Prompt_j, \theta) \tag{4}$$

We use a quantized version of the model due to memory limitations and perform Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2024) as a parameter-efficient fine-tuning technique. The training details are provided in the Section 4.

3.2.2 Cause prediction

To prepare the dataset for cause prediction, we incorporate the emotion labels obtained for each utterance. The conversation context now includes the emotion labels for each utterance u_j excluding those with a predicted emotion label $\hat{y_j^e}$ of *neutral*. This approach enhances the model's ability to analyze causal dependencies and identify which utterances may have contributed to a specific emotion. The output for cause prediction is a list of cause utterance IDs. The instruction is provided in Appendix E.1. The modified prompt for this step consists of this instruction I_j^c along with the conversational context with emotion labels C_j^e:

$$Prompt_j = (C_j^e, I_j^c) \tag{5}$$

Next, we perform supervised fine-tuning of a new Llama 2-13b model using this prompt as the input and the corresponding true list of causes:

$$\hat{y_j^c} = \textbf{Llama}_c(Prompt_j, \theta) \tag{6}$$

3.2.3 Adding video captions

To integrate cues from the videos corresponding to each utterance, we experimented using video captions generated using GPT-4 Vision as additional context for the model. However, we observed a notable decrease in performance since descriptions for individual utterances were somewhat noisy and did not effectively guide the predictions. Moreover, the captions often contained multiple emotions causing confusion for the model. As a result, we do not utilize these during training.

3.3 Approach 2: In-Context-Learning GPT

Our second approach (Fig. 4) tackles subtask 2 by obtaining conversation-level video captions using the GPT-4V(ision) model by OpenAI (Yang

[1]https://huggingface.co/meta-llama/Llama-2-13b-chat-hf
[2]https://www.llama2.ai/

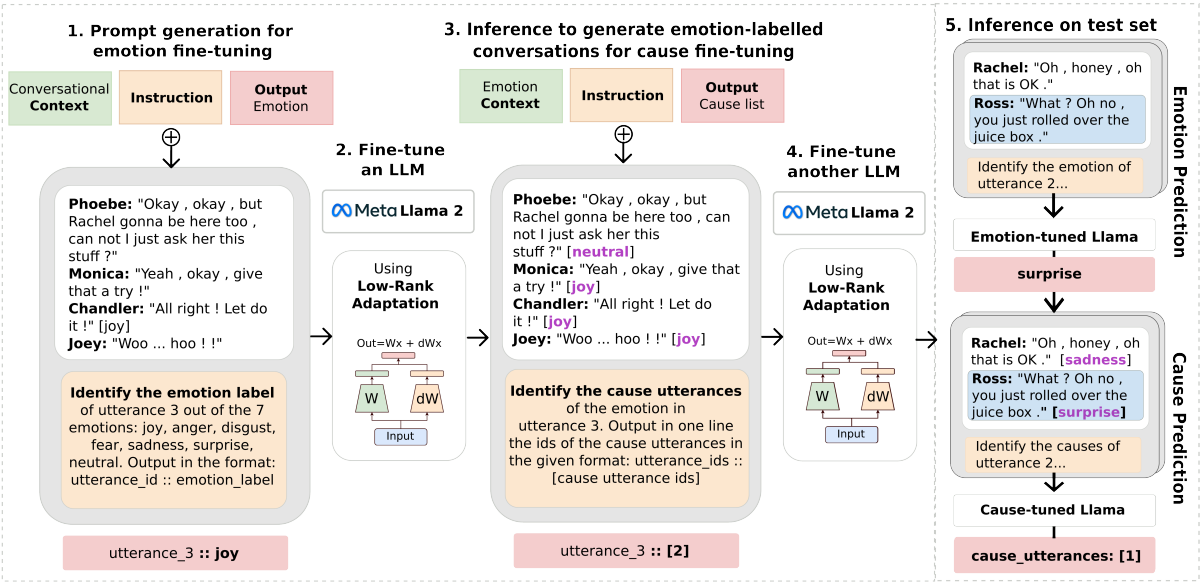


Figure 3: Pipeline for fine-tuning Llama (Approach 1)

et al., 2023). For emotion prediction, we retrieve a semantically similar conversation from the training set whose emotion annotations are explained as demonstration examples in the prompt for the GPT-3.5 model[3]. For each predicted emotional utterance, we perform cause prediction within a context window around the emotional utterance. Due to the complex nature of the task, we leverage in-context-learning (Dong et al., 2023) by retrieving similar context windows from the training set whose cause annotations are explained as demonstration examples in the prompt for the GPT-3.5 model. We discuss each step in the subsequent sections.

3.3.1 Video Captioning

GPT-4V has the capability to process video sequences (Yang et al., 2023; Lin et al., 2023). In our approach, we extract conversation-level captions from the videos. However, due to rate limits and the costs considerations, we use a compact image representation for each video associated with the utterances of a conversation. Therefore, these image sequences serve as input to the GPT-4V model, generating a description for the entire conversation. The prompt is shown in the Fig. 5.

For an utterance, we sample nine equidistant frames across its video length. These frames aim to capture the dynamics of the whole video. We arrange these frames in a 3×3 grid, following a row-major order. Additionally, we include the speaker text below the grid to provide further context to GPT-4V. The process is illustrated in Fig. 5.

To accommodate the rate limits of the Vision API, we batch the utterances of a conversation and obtain outputs independently from the Vision model. We stitch all the outputs of a batched conversation into a single caption using GPT-3.5 (Appendix Fig. 16).

3.3.2 Emotion Recognition

GPT tends to be uncontrollable when performing zero-shot recognition of emotions in conversations (Qin et al., 2023) outputting emotions that are not a valid category of labels. To guide and control the process, we leverage in-context learning (ICL) by retrieving a conversation from the training set whose emotions are already annotated. The emotions in these conversations are explained by GPT-3.5 (Appendix Fig. 17). This retrieved conversation and its explanation serve as a demonstration for GPT to learn from, enabling it to recognize emotions in conversations more accurately. In addition, the prompt template includes the video caption as part of the input, as shown in Appendix Fig. 18.

To ensure effective ICL, it is important to provide general and descriptive examples that aid in solving the current task. In our approach, we sampled conversations from the training set containing all emotion categories. These conversations were stored as text-embedding-ada-002 embeddings (Neelakantan et al., 2022) in a vector database. At test time, we compute the embedding

[3]https://platform.openai.com/docs/models/gpt-3-5-turbo

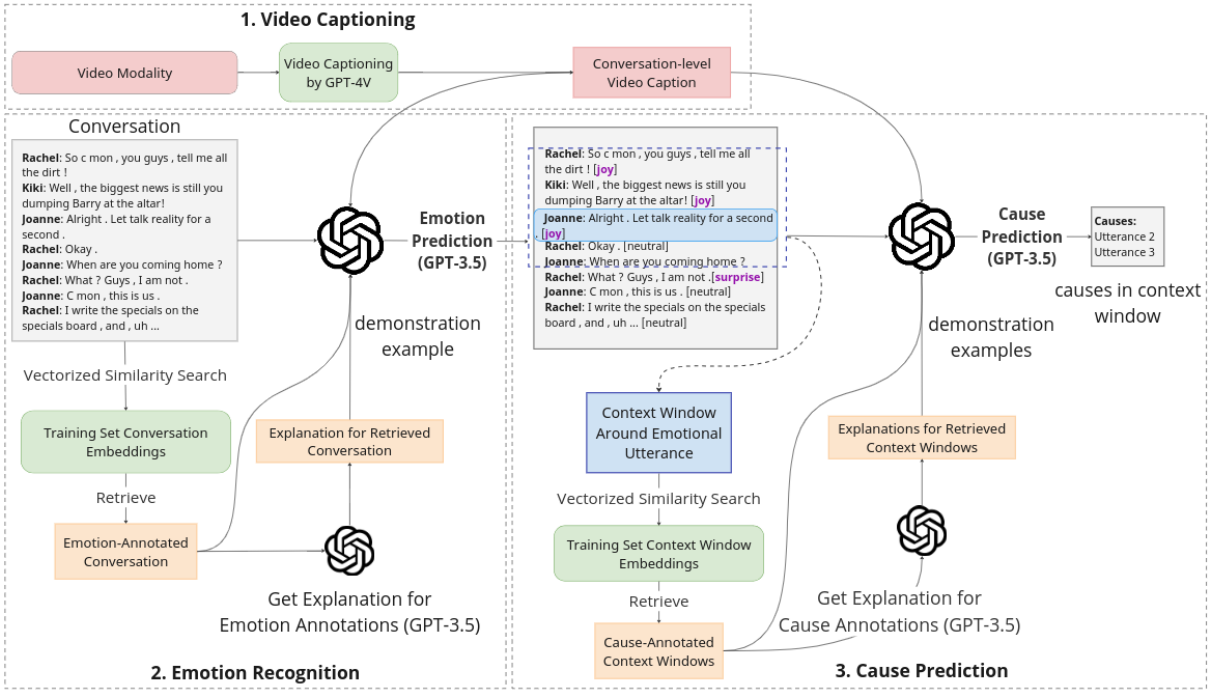


Figure 4: Pipeline of In-Context-Learning GPT Method (Approach 2)

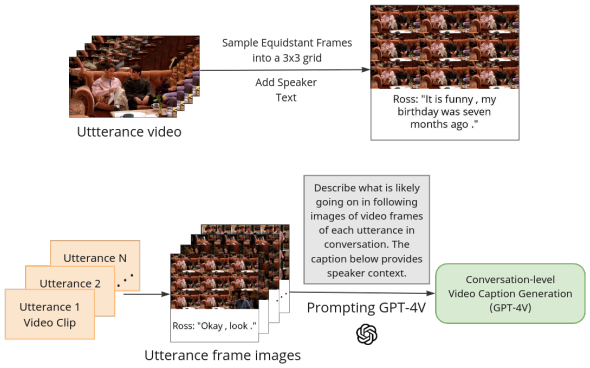


Figure 5: Video Captioning Pipeline

for a conversation and retrieve the closest matching embedding from the database based on Euclidean distance. The retrieved embedding aids ICL in improving emotion understanding and recognition.

3.3.3 Cause Prediction

Following the prediction of emotions, we predict the causes for each emotional utterance within a context window around that utterance. The bounds of the context window are given in Table 2. The bounds were informed by the distribution of the majority of relative positions of causes in the training set (Figure 2).

For predicting the causes of an utterance with emotion e within a given context window c, we retrieve context windows containing utterances with the same emotion e that exhibit semantic similarity to c. This retrieval is accomplished through the Euclidean distance comparison of text-embedding-ada-002 embeddings derived from the training data. The retrieved conversation's causes are explained by GPT-3.5 (Appendix Fig. 19). Learning from the explained retrieved-context windows, cause prediction on c can be performed by GPT-3.5. Video captions are also included in the prompt (Appendix Fig. 20), since the local window may have lost some broader context.

Position	Previous	Next
Beginning	0	2
End	5	0
Middle	5	2

Table 2: Context Window Bounds in each Direction

3.4 Post-Processing

In both our approaches, after getting the causes, we perform a post-processing step where we add the emotional utterance as its own cause which we call self-causes. This gives significant performance boosts as a majority of the causes are self-causes as pointed out in Appendix 2.3.

4 Experimental setup

Training details For approach 1, the data is split into train, test, and validation sets in the ratio 8:1:1.

We use peft library [4] for Parameter-Efficient Fine-Tuning. Due to memory constraints, we fine-tune a 4-bit quantized Llama-2 model using bitsandbytes library [5]. We report the details of the implementation for both approaches in Appendix B.

Evaluation metrics For evaluating, we report the precision, recall, F1-score, and weighted F1 which can be found on the competition website.[6]

5 Results and Discussion

Main results Both of our approaches gave competitive rankings on the official leaderboard for subtask 2 as shown in Table 3. In-context-learning GPT gave better results on the evaluation set compared to Fine-tuned Llama, thus our final position on the leaderboard was rank 4.

System	w-avg F1	F1
1. Samsung Research China-Beijing	0.3774	0.3870
2. NUS-Emo	0.3460	0.3517
3. SZTU-MIPS	0.3435	0.3434
4. GPT-ICL *(Ours)*	**0.2758**	**0.2816**
5. MotoMoto	0.2584	0.2595
6. Fine-tuned Llama *(Ours)*	**0.2558**	**0.2630**

Table 3: Leaderboard Results on Evaluation Data

Ablation study We conduct extensive ablation studies to measure the importance of the techniques we employ summarized in Table 4. For these experiments, we use a subset of our test set containing 528 utterances. It can be seen that the performance of zero-shot Llama as well as GPT is the lowest. Instruction-tuning and ICL clearly improve the performance on the task, showcasing the significance of making LLMs context-aware when tackling downstream tasks. Adding self-causes improves performance in both zero-shot and context-aware cases highlighting their importance. The incorporation of video captions leads to poorer results in context-learning. The detailed table is in Appendix C.

Limitations Our approaches are specific to one dataset and may not generalize well to other datasets. Due to resource limitations, we fine-tune a Llama 13b parameter model instead of 70b and use QLoRA instead of updating all parameters. To save costs, we used GPT-3.5 model instead of GPT-4. Even with extensive prompt engineering, GPT models tend to hallucinate or give unstructured outputs, requiring retry repeatedly.

Approach	F1	w-avg F1
Zero-shot Llama		
- w/o self-causes	0.117	0.116
- w/ self-causes	0.222	0.215
Instruction-tuned Llama		
- w/o self-causes	0.325	0.318
- w/ self-causes	**0.364**	**0.352**
Zero-shot GPT		
- w/o self-causes	0.100	0.097
- w/ self-causes	0.189	0.184
In-context-learning GPT		
- w/o self-causes w/o video	0.286	0.296
- w/o self-causes w/ video	0.235	0.241
-w/ self-causes w/o video	**0.336**	**0.342**
-w/ self-causes w/ video	0.329	0.334

Table 4: Results on Validation Set.

6 Conclusion

We tackled the Multimodal ECAC task with a two-step framework of recognizing emotions first and then predicting their causes using LLMs. We implemented two approaches: a Llama-2 model which has been fine-tuned with instructions and a GPT model which solves the task by learning from demonstration examples in context. Conversation-level video captions were extracted to provide more context to LLMs. Our second approach was our best submission for the task, placing us at rank 4 with our first approach being placed at rank 6. Our results were under cost constraints and further investigation with larger Llama-2 models and GPT-4 with more sophisticated ICL approaches are a clear follow-up of our work.

References

Sharmeen M Saleem Abdullah Abdullah, Siddeeq Y Ameen Ameen, Mohammed AM Sadeeq, and Subhi Zeebaree. 2021. Multimodal emotion recognition using deep learning. *Journal of Applied Science and Technology Trends*, 2(02):52–58.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Pablo Barros, Nikhil Churamani, Egor Lakomkin, Henrique Siqueira, Alexander Sutherland, and Stefan

[4]https://huggingface.co/docs/peft/en/index
[5]https://github.com/TimDettmers/bitsandbytes
[6]https://nustm.github.io/SemEval-2024_ECAC/

Wermter. 2018. The omg-emotion behavior dataset. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Ying Chen, Wenjun Hou, Shoushan Li, Caicong Wu, and Xiaoqiang Zhang. 2020. End-to-end emotion-cause pair extraction with graph convolutional network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 198–207.

Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 179–187.

Huang-Cheng Chou, Wei-Cheng Lin, Lien-Chiang Chang, Chyi-Chang Li, Hsi-Pin Ma, and Chi-Chun Lee. 2017. Nnime: The nthu-ntua chinese interactive multimodal emotion corpus. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 292–298. IEEE.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *ArXiv*, abs/2301.00234.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Chuang Fan, Chaofa Yuan, Jiachen Du, Lin Gui, Min Yang, and Ruifeng Xu. 2020. Transition-based directed graph construction for emotion-cause pair extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3707–3717.

Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Meisd: A multimodal multi-label emotion, intensity and sentiment dialogue dataset for emotion recognition and sentiment analysis in conversations. In *Proceedings of the 28th international conference on computational linguistics*, pages 4441–4453.

Yao Fu, Shaoyang Yuan, Chi Zhang, and Juan Cao. 2023. Emotion recognition in conversations: A survey focusing on context, speaker dependencies, and fusion methods. *Electronics*, 12(22):4714.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part II 16*, pages 152–165. Springer.

Lin Gui, Ruifeng Xu, Dongyin Wu, Qin Lu, and Yu Zhou. 2018. Event-driven emotion cause extraction with corpus construction. In *Social Media Content Analysis: Natural Language Processing and Beyond*, pages 145–160. World Scientific.

Chao-Chun Hsu, Sheng-Yeh Chen, Chuan-Chun Kuo, Ting-Hao Huang, and Lun-Wei Ku. 2018. Emotionlines: An emotion corpus of multi-party conversations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Guimin Hu, Guangming Lu, and Yi Zhao. 2021. Fss-gcn: A graph convolutional networks with fusion of semantic and structure for emotion cause analysis. *Knowledge-Based Systems*, 212:106584.

Mia Mohammad Imran, Preetha Chatterjee, and Kostadin Damevski. 2023. Uncovering the causes of emotions in software developer communication using zero-shot llms. *arXiv preprint arXiv:2312.09731*.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework.

Weiyuan Li and Hua Xu. 2014. Text-based emotion classification using emotion cause extraction. *Expert Systems with Applications*, 41(4):1742–1749.

Xiangju Li, Wei Gao, Shi Feng, Yifei Zhang, and Daling Wang. 2021. Boundary detection with bert for span-level emotion cause analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 676–682.

Yong Li, Yuanzhi Wang, and Zhen Cui. 2023. Decoupled multimodal distilling for emotion recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6631–6640.

Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, Ce Liu, and Lijuan Wang. 2023. Mm-vid: Advancing video understanding with gpt-4v(ision). *ArXiv*, abs/2310.19773.

Trisha Mittal, Puneet Mathur, Aniket Bera, and Dinesh Manocha. 2021. Affect2mm: Affective analysis of multimedia content using emotion causality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5661–5671.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al.

2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Sancheng Peng, Lihong Cao, Yongmei Zhou, Zhouhao Ouyang, Aimin Yang, Xinguang Li, Weijia Jia, and Shui Yu. 2022. A survey on deep learning for textual emotion analysis in social networks. *Digital Communications and Networks*, 8(5):745–762.

Patrícia Pereira, Helena Moniz, and Joao Paulo Carvalho. 2022. Deep emotion recognition in textual conversations: A survey. *arXiv preprint arXiv:2211.09172*.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Nicu Sebe, Ira Cohen, Theo Gevers, and Thomas S Huang. 2005. Multimodal approaches for emotion recognition: a survey. In *Internet Imaging VI*, volume 5670, pages 56–67. SPIE.

Nusrat J Shoumy, Li-Minn Ang, Kah Phooi Seng, DM Motiur Rahaman, and Tanveer Zia. 2020. Multimodal big data affective analytics: A comprehensive survey using text, audio, visual and physiological signals. *Journal of Network and Computer Applications*, 149:102447.

Aaditya Singh, Shreeshail Hingane, Saim Wani, and Ashutosh Modi. 2021. An end-to-end network for emotion-cause pair extraction. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 84–91.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023a. Multimodal emotion-cause pair extraction in conversations. *IEEE Trans. Affect. Comput.*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024a. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. 2024b. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36.

Yubo Wang, Xueguang Ma, and Wenhu Chen. 2023b. Augmenting black-box llms with medical textbooks for clinical question answering.

Yuwei Wang, Yuling Li, Kui Yu, and Yimin Hu. 2023c. Knowledge-enhanced hierarchical transformers for emotion-cause pair extraction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 112–123. Springer.

Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023d. Is chatgpt a good sentiment analyzer? a preliminary study. *arXiv preprint arXiv:2304.04339*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Penghui Wei, Jiahao Zhao, and Wenji Mao. 2020. Effective inter-clause modeling for end-to-end emotion-cause pair extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3171–3181.

Martin Wöllmer, Felix Weninger, Tobias Knaup, Björn Schuller, Congkai Sun, Kenji Sagae, and Louis-Philippe Morency. 2013. Youtube movie reviews: Sentiment analysis in an audio-visual context. *IEEE Intelligent Systems*, 28(3):46–53.

Jialiang Wu, Yi Shen, Ziheng Zhang, and Longjun Cai. 2024. Enhancing large language model with decomposed reasoning for emotion cause pair extraction. *arXiv preprint arXiv:2401.17716*.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Shuntaro Yada, Kazushi Ikeda, Keiichiro Hoashi, and Kyo Kageura. 2017. A bootstrap method for automatic rule acquisition on emotion cause extraction. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 414–421. IEEE.

Yi Yang, Qingwen Zhang, Ci Li, Daniel Simões Marta, Nazre Batool, and John Folkesson. 2024. Human-centric autonomous systems with llms for user command reasoning. In *Proceedings of the IEEE/CVF*

Winter Conference on Applications of Computer Vision (WACV) Workshops, pages 988–994.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v(ision). *ArXiv*, abs/2309.17421.

Wenmeng Yu, Hua Xu, Fanyang Meng, Yilin Zhu, Yixiao Ma, Jiele Wu, Jiyun Zou, and Kaicheng Yang. 2020. Ch-sims: A chinese multimodal sentiment analysis dataset with fine-grained annotation of modality. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3718–3727.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*.

Shiqing Zhang, Yijiao Yang, Chen Chen, Xingnan Zhang, Qingming Leng, and Xiaoming Zhao. 2023. Deep learning-based multimodal emotion recognition from audio, visual, and text modalities: A systematic review of recent advancements and future prospects. *Expert Systems with Applications*, page 121692.

Xiaoheng Zhang and Yang Li. 2023. A cross-modality context fusion and semantic refinement network for emotion recognition in conversation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13099–13110.

Yazhou Zhang, Mengyao Wang, Youxi Wu, Prayag Tiwari, Qiuchi Li, Benyou Wang, and Jing Qin. 2024. Dialoguellm: Context and emotion knowledge-tuned large language models for emotion recognition in conversations.

Weixiang Zhao, Yanyan Zhao, Zhuojun Li, and Bing Qin. 2023. Knowledge-bridged causal interaction network for causal emotion entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14020–14028.

Li Zheng, Donghong Ji, Fei Li, Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, and Chong Teng. 2023a. Ecqed: Emotion-cause quadruple extraction in dialogs. *arXiv preprint arXiv:2306.03969*.

Wenjie Zheng, Jianfei Yu, Rui Xia, and Shijin Wang. 2023b. A facial expression-aware multimodal multi-task learning framework for emotion recognition in multi-party conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15445–15459.

Xiaopeng Zheng, Zhiyue Liu, Zizhen Zhang, Zhaoyang Wang, and Jiahai Wang. 2022. Ueca-prompt: Universal prompt for emotion cause analysis. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7031–7041.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 165–176.

A Related Work

Our system is designed to prioritize Subtask 2 which is directly related to text-based and multimodal ECA. In the following sections, we will present relevant research that addresses both unimodal (text-based) and multimodal ECA.

Text-based ECA

Advancements in text-based ECA (Xia and Ding, 2019; Hsu et al., 2018; Peng et al., 2022; Pereira et al., 2022) have made significant strides within the field of sentiment analysis. The task on emotion cause extraction (ECE) was initially proposed by Chen et al. (2010) on a Chinese corpus. Several studies (Li and Xu, 2014; Ghazi et al., 2015; Yada et al., 2017) have explored ECE task, using both rule-based and machine learning approaches that operate at the phrase or word level of the text data. Furthermore, (Gui et al., 2018) reformulated the ECE task as a clause-level classification problem and constructed a Chinese emotion-cause corpus based on the news data. Considering the effectiveness of clause-level units in indicating emotions, Xia and Ding (2019) introduced the task of Emotion-Cause Pair Extraction (ECPE) for extracting potential emotion-cause pairs from texts. Numerous deep learning models (Zhong et al., 2019; Wei et al., 2020; Chen et al., 2020; Singh et al., 2021; Li et al., 2021; Wang et al., 2023c) have been developed to address ECPE tasks. Additionally, graph-based approaches (Zheng et al., 2023a; Hu et al., 2021; Zhao et al., 2023) that utilize graphs to model dialog context and capture interactions between speakers and utterances hold significant potential. The focus on transformer models and the rapid progress in LLMs such as ChatGPT[7] and Llama (Touvron et al., 2023), have significantly boosted the performance of various NLP tasks (Imran et al., 2023) including ECPE (Wang et al., 2023d; Imran et al., 2023; Wu et al., 2024; Zheng et al., 2022).

[7] https://chat.openai.com/

Multimodal ECA

Given the strong association between facial cues and emotion, integrating modalities to improve emotion recognition has attracted a lot of attention (Sebe et al., 2005; Li et al., 2023; Zhang et al., 2023; Fu et al., 2023). Several key multimodal datasets (Wöllmer et al., 2013; Zadeh et al., 2016; Chou et al., 2017; Barros et al., 2018; Poria et al., 2019; Yu et al., 2020) have emerged to support and advance research. The availability of open conversation data has facilitated the expansion of multimodal conversation datasets, which includes various types of conversations such as dyadic interactions (Busso et al., 2008), and multi-participant communications (Hsu et al., 2018; Poria et al., 2019; Firdaus et al., 2020; Zheng et al., 2023b).

Large Language Models

The emergence of Large Language Models such as GPT-4 (Achiam et al., 2023), Llama (Touvron et al., 2023), PaLM (Anil et al., 2023), etc. has transformed the research landscape. Recently, there has been a surge in the application of LLMs to a multitude of domains. Zhang et al. (2024) extend their capabilities to the task of emotion recognition where they fine-tune a Llama 2-7 billion parameter model for emotion prediction. Lei et al. (2023) introduce a retrieval template module along with speaker identification and emotion-impact prediction tasks to improve the performance of LLM. In our work, as part of approach 1, we develop two distinct LLM-based experts separately for emotion and cause prediction.

Qin et al. (2023) investigated the task of zero-shot emotion cause prediction using ChatGPT with limited success. Recently, a new paradigm of in-context learning (ICL) (Dong et al., 2023) has emerged for LLMs that involves learning from a few examples to solve a variety of complex reasoning tasks (Wei et al., 2022b), (Wei et al., 2022a). Wu et al. (2024) proposed a Chain of Thought (CoT) (Wei et al., 2022b) approach for emotion cause pair extraction. Our approach 2 extends the idea of ICL towards solving the task of multimodal emotion cause pair extraction in two steps.

B Implementation details

B.1 Training details for Llama

Both emotion and cause prediction training used one Nvidia A100 40GB GPU for training (Available on Google Colab Pro priced at $11.8/month).

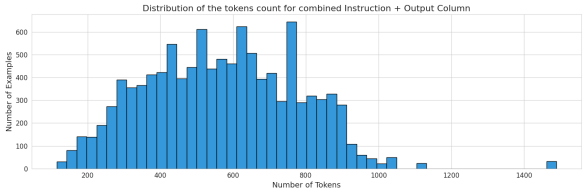


Figure 6: Distribution of token counts for Llama tokenizer

We train for one epoch due to constraints on Colab usage with gradient accumulation steps as 8 with an effective batch size of 8. A cosine learning rate scheduler and Adam optimizer are used. Inference is performed using two Tesla T4 16GB GPUs (Available on Kaggle for free (30 hrs/month)).

The long context length of 4096 tokens of the Llama 2 models, allows us to include the entire conversation as context and input that to the model. We perform experiments to analyze the maximum token counts in the dataset and observe that they do not exceed 1600 as shown in Figure 6. In case the token count exceeds the limit for the LLM we can use a window of utterances around the given utterance as context for predicting its emotion.

Hyperparameter	Value
Lora alpha	16
Lora dropout	0.1
Attention heads	16
Learning rate	1e-3
Epochs	1
LR scheduler	cosine
Warmup ratio	0.03
Weight decay	0.001

Table 5: Hyperparameters for fine-tuning

B.2 Details for in-context learning GPT

We use the LangChain[8] library to implement our three pipelines: video captioning, emotion recognition, and cause prediction. We use the interface provided by LangChain to communicate with OpenAI's API models detailed in Table 6.

Model	API Name
GPT-4V	gpt-4-vision-preview
GPT-3.5	gpt-3.5-turbo-1106
Embeddings	text-embedding-ada-002

Table 6: OpenAI API Model Names

Vector databases For creating vector databases, we use the FAISS Library (Douze et al., 2024). We

[8]https://github.com/langchain-ai/langchain

created a FAISS index containing embeddings of 12 conversations from the training set which contains all emotion categories. For cause prediction, we created a FAISS index for each of the 6 emotion categories and 3 possible positions of emotional utterance giving us a total of 18 indices. Each of these indices contained embeddings of context windows (bounds defined in Table 2) from the training set corresponding to each emotion and position.

C Detailed results

The detailed results on precision, recall, and F1-scores are given in Table 7.

D Error Analysis

We conduct error analysis for the output of emotion recognition using the two approaches. The performance of zero-shot Llama is extremely poor where the model predicts the label joy for almost all utterances (Fig. 7). On adding the conversational context, the model can identify the emotional nuances better, yet often predicts joy or surprise for neutral (Fig. 8). Instruction fine-tuning significantly boosts performance where the model can now differentiate distinct emotions (Fig. 9). The performance on disgust and fear is low due to the class-imbalance problem. In our test subset, the support of disgust and fear is only 13, as shown in Table 8. We observed similar trends in the case of our second approach. Zero-shot GPT (Fig. 10) tends to only identify the neutral utterances accurately and fails in other categories. The incorporation of in-context learning (Fig. 11) improves the accuracy in identifying different emotion categories but there is little to no improvement in identifying disgust or anger utterances.

E Prompt details

E.1 Fine-tuned Llama 2

The general prompt for the Llama chat version is given in Figure 14. The prompts for emotion and cause prediction are given in Fig. 12 and Fig. 13. We provide a specific format for the output so as to ease the post-processing where we extract the first emotion label occurring after the "::" sequence of characters.

E.2 ICL-GPT

We devise prompt templates to be used in the LangChain framework. {} represent placeholders to be replaced when making a prompt. Video

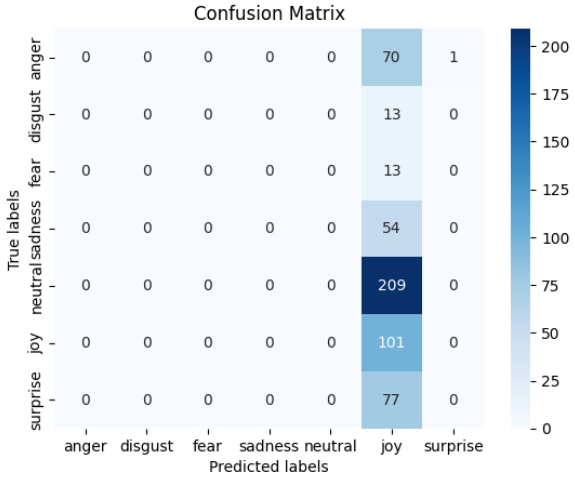


Figure 7: Confusion matrix for zero-shot emotion recognition without context using Llama

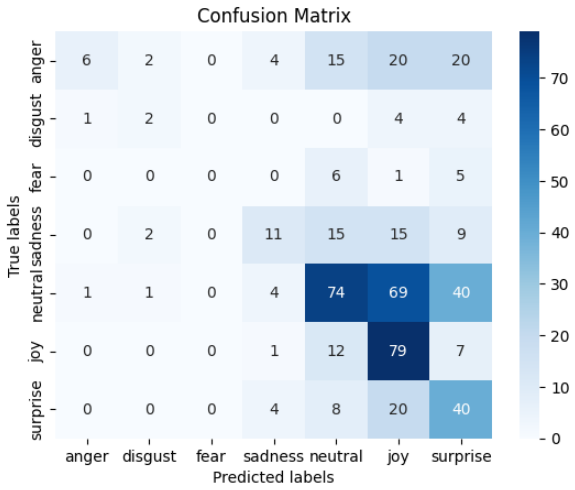


Figure 8: Confusion matrix for zero-shot emotion recognition with context using Llama

captioning prompt is given in Fig. 15. Due to rate limits, we had to batch the utterances, thus we may have multiple disjoint descriptions of a conversation. We prompt GPT-3.5 using the prompt in Fig. 16 to stitch the descriptions into a single caption. For explaining the retrieved conversation with emotion annotated, we use the prompt in Fig. 17. The retrieved conversation and explanation are now used as demonstration examples for the emotion recognition prompt in Fig. 18. For an explanation of causes in the retrieved-context window, we use the prompt in Fig. 19. The explanations of the retrieved windows are used as demonstration examples in the prompt for cause prediction within a context window as shown in the prompt in Fig. 20.

Approach	P	R	F1	w-P	w-R	w-avg F1
Zero-shot Llama w/o self-causes	0.089	0.168	0.117	0.090	0.168	0.116
Zero-shot Llama w/ self-causes	0.157	0.372	0.222	0.152	0.372	0.215
Instruction-tuned Llama w/o self-causes	0.351	0.304	0.325	0.335	0.304	0.318
Instruction-tuned Llama w/ self-causes	**0.360**	**0.367**	**0.364**	**0.342**	**0.367**	**0.352**
Zero-shot GPT w/o self-causes	0.081	0.130	0.100	0.087	0.130	0.097
Zero-shot GPT w/ self-causes	0.140	0.290	0.189	0.149	0.290	0.184
In-context-learning GPT w/o video captions w/o self-causes	0.259	0.319	0.286	0.283	0.319	0.296
In-context-learning GPT w/o video captions w/ self-causes	**0.270**	**0.445**	**0.336**	**0.287**	**0.445**	**0.342**
In-context-learning GPT w/o self-causes	0.216	0.256	0.235	0.241	0.256	0.241
In-context-learning GPT w/ self-causes	0.261	0.445	0.329	0.280	0.445	0.334

Table 7: Results on Validation Set. P: precision, R: recall, w: weighted.

Approach	Metric Supp	Anger 71	Disgust 13	Fear 13	Joy 101	Sadness 54	Surprise 77	Neutral 209
Zero-shot Llama w/o context	**P**	0.0000	0.0000	0.0000	0.1881	0.0000	0.0000	0.0000
	R	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
	F1	0.0000	0.0000	0.0000	0.3166	0.0000	0.0000	0.0000
Zero-shot Llama with context	**P**	0.7500	0.2857	0.0000	0.3798	0.4583	0.3200	0.5663
	R	0.0845	0.1538	0.0000	0.7822	0.2037	0.5195	0.4498
	F1	0.1519	**0.2000**	0.0000	0.5113	0.2821	0.3960	0.5013
Fine-tuned Llama with context	**P**	0.5641	0.0	0.3333	0.6210	0.625	0.6103	0.6666
	R	0.6197	0.0	0.1538	0.5842	0.3704	0.6104	0.7943
	F1	**0.5906**	0.0	**0.2105**	**0.6020**	**0.4651**	**0.6104**	**0.7249**
Zero-Shot GPT	**P**	0.5652	0.2500	0.2727	0.4265	0.5385	0.5200	0.5906
	R	0.3333	0.4000	0.4286	0.5370	0.1842	0.3023	0.7426
	F1	0.4194	0.3077	0.3333	0.4754	0.2745	0.3824	0.6580
In-Context-Learning GPT	**P**	0.6667	0.2222	0.2222	0.4595	0.7000	0.5610	0.6957
	R	0.4615	**0.4000**	**0.2857**	0.6296	0.3684	0.5349	0.7059
	F1	**0.5455**	0.2857	0.2500	**0.5312**	**0.4828**	**0.5476**	**0.7007**

Table 8: Emotion Recognition Results for Seven Emotion Categories. P: precision, R: recall, F1: F1 score, and Supp: support.

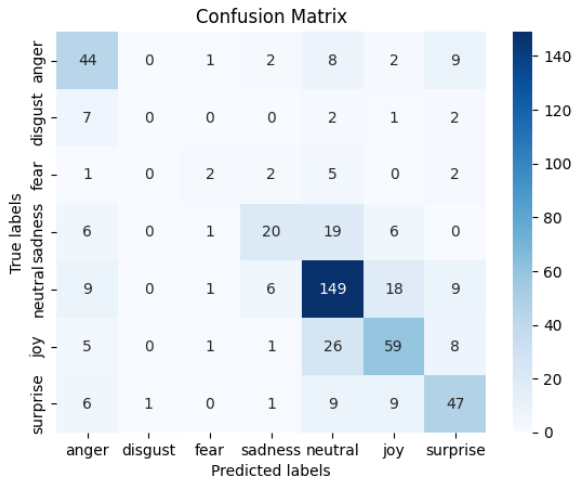


Figure 9: Confusion matrix for emotion recognition with context using fine-tuned Llama

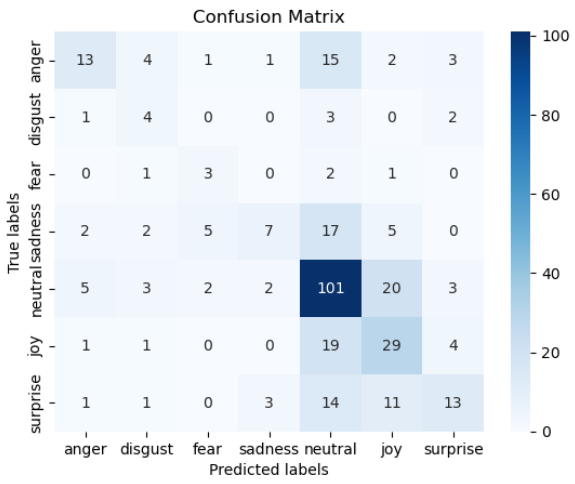


Figure 10: Confusion matrix for emotion recognition using Zero-shot GPT-3.5

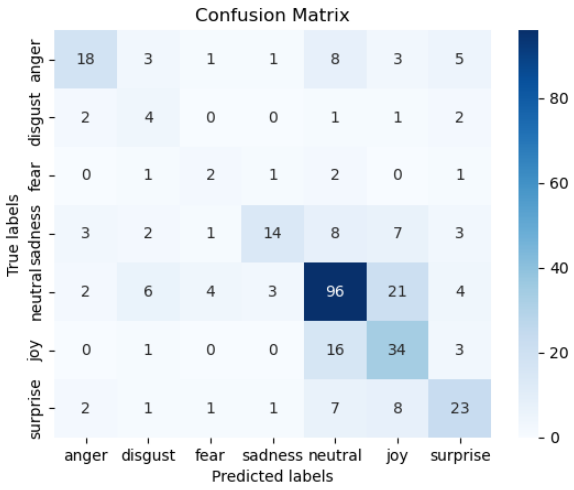


Figure 11: Confusion matrix for emotion recognition using GPT-ICL

Emotion Recognition Prompt:

<s>[INST]
"conversation": [
{
"utterance_ID": 1,
"text": "This is just Bactine . It will not hurt .",
"speaker": "Monica"
},
{
"utterance_ID": 2,
"text": "Sorry , that was wax .",
"speaker": "Joey"
},
{
"utterance_ID": 3,
"text": "Oh , poor little Tooty is scared to death . We should find his owner .",
"speaker": "Phoebe"
},
{
"utterance_ID": 4,
"text": "Why do not we just put poor little Tooty out in the hall ?",
"speaker": "Ross"
},
{
"utterance_ID": 5,
"text": "During a blackout ? He would get trampled !",
"speaker": "Rachel"
}]
Identify the emotion label of utterance 5 out of the 7 emotions: anger, fear, disgust, sadness, joy, surprise, neutral. Don't give an explanation.
Output only one line in the format:
utterance_id :: emotion_label
[/INST]
utterance_5 :: anger
</s>

Figure 12: Example Prompt for emotion prediction using Llama

Cause Prediction Prompt:
<s>[INST]
"conversation": [
{
"utterance_ID": 1,
"text": "This is just Bactine . It will not hurt .",
"speaker": "Monica",
"emotion": "neutral"
},
{
"utterance_ID": 2,
"text": "Sorry , that was wax .",
"speaker": "Joey",
"emotion": "neutral"
},
{
"utterance_ID": 3,
"text": "Oh , poor little Tooty is scared to death . We should find his owner .",
"speaker": "Phoebe",
"emotion": "sadness"
},
{
"utterance_ID": 4,
"text": "Why do not we just put poor little Tooty out in the hall ?",
"speaker": "Ross",
"emotion": "disgust"
},
{
"utterance_ID": 5,
"text": "During a blackout ? He would get trampled !",
"speaker": "Rachel",
"emotion": "anger"
}]
Identify the cause utterances of the emotion in utterance 5. Output in one line the ids of the cause utterances as a list in the given format:
utterance_id :: [cause utterance ids]
Don't give any explanation.
[/INST]
utterance_id :: [4,5]
</s>

Figure 13: Example Prompt for cause prediction using Llama

General Prompt Template:
\<s>[INST] <<SYS>>
{{system message}}

<</SYS>>

{{message/input}}
[/INST]
{{answer}}
</s>

Figure 14: General Prompt Template for Llama

Video Caption Prompt:
You are an expert of Friends TV Show. You can understand a video scene from a few of its frames shown in sequence. You give precise descriptive analysis.
Describe what is likely going on in following images of video frames of each utterance in conversation. The caption below provides speaker context.
Give output as:
Scene Description:

Figure 15: Video Captioning Prompt Template

Caption Stitching Prompt:
Following is a descriptions of video clip from Friends TV show for a particular conversation. The descriptions are broken from each other. Stitch the description into a continous coherent narrative of the whole scene

Figure 16: Batched Video Caption Stitching Prompt Template

Emotion Explanation Prompt:
There are 6 basic emotions: Anger, Disgust, Fear, Joy, Sadness, Surprise.
The emotion of the speaker is determined by the context of the conversation.
If the emotion is not in any category, is a mix of several categories, or is ambiguous it can be categorized as "Neutral".
Analyze the following conversation where emotion of each utterance is annotated in square brackets at the end.
Give reasoning behind the annotation of each utterance.

{conversation}

Output a JSON in the following format:
[{{"utterance_ID": id,
"text" : content,
"speaker": speaker
"emotion": emotion,
"explanation": detailed explanation}}
...
]
No plain text.

Figure 17: Emotion Label Explanation Prompt Template

Emotion Recognition Prompt:
You are a die-hard fan of the popular Friends TV show.
You have all the knowledge of all the seasons and are familiar with all the characters.
Your task is to recognize emotions in utterances.
Here's an annotated example with recognized emotion and explanation:
{example}

Like above example annotate the following Conversation:
Context for the scene is given below:
{scene}

Conversation:
{conversation}

Classify the emotional state of the speaker in each utterance into ONLY one out of the 6 emotions:
Anger, Disgust, Fear, Joy, Sadness, Surprise.
The emotion of the speaker is determined by the context of the conversation.
Give explanation for your classification using the context. Only Use the above 6 emotion categories.
If the emotion is not in any category, is a mix of several categories, or is ambiguous,
classify the state as "Neutral". Sarcastic comments may be categorized as Neutral.
Format the output as JSON as the given example. No plain text.

Figure 18: Emotion Recognition with Context Learning Prompt Template

Cause Explanation Prompt:
You are an expert in analyzing conversations to extract the causes of emotions
in particular utterances by speakers. You give definite confident answers only.
Description of emotional causes:
- Each utterance always has a reason of why it was said and why it had a particular emotion.
- A cause is an utterance that comes before or after the particular utterance in question that best explains to be the reason behind the particular emotion.
- The emotional utterance itself can be a cause of itself if its content ALSO best explains the reason for the particular emotion.
- Sometimes the cause can be beyond the context of the conversation thus an utterance might have no cause within conversation
- There can be multiple causes for an utterance.

Here's a conversation:
{conversation}

Analyze and justify the above annotation concisely.

Figure 19: Cause Explanations Prompt Template

Cause Prediction Prompt:
You are an expert in analyzing conversations to extract the causes of emotions
in particular utterances by speakers. You give definite confident answers only.
Description of emotional causes:
- Each utterance always has a reason of why it was said and why it had a particular emotion.
- A cause is an utterance that comes before or after the particular utterance in question that best explains to be the reason behind the particular emotion.
- The emotional utterance itself can be a cause of itself if its content ALSO best explains the reason for the particular emotion.
- Sometimes the cause can be beyond the context of the conversation thus an utterance might have no cause within conversation
- There can be multiple causes for an utterance.

Here are some examples of how to recgonize causes:
Example 1:
{example_1}

Example 2:
{example_2}

Example 3:
{example_3}

Now, please recognize the causes in following conversation. Heres the context for the whole conversation:
{scene}

Conversation:
{window}

Figure 20: Cause Prediction with Context Learning Prompt Template

LMU-BioNLP at SemEval-2024 Task 2: Large Diverse Ensembles for Robust Clinical NLI

Zihang Sun*, **Danqi Yan***, **Anyi Wang***, **Tanalp Agustoslu***, **Qi Feng***, **Chengzhi Hu***, **Longfei Zuo***, **Shijia Zhou***, **Hermine Kleiner***, **Pingjun Hong***, **Suteera Seeha***, **Sebastian Loftus***, **Anna Susanna Barwig***, **Oliver Kraus***, **Jona Volohonsky***, **Yang Sun***, **Leopold Martin***, **Lena Altinger***, **Jing Wang***, **Leon Weber-Genzel**
LMU Munich, Germany

leonweber@cis.lmu.de

Abstract

In this paper, we describe our submission for the NLI4CT 2024 shared task on robust Natural Language Inference over clinical trial reports. Our system is an ensemble of nine diverse models which we aggregate via majority voting. The models use a large spectrum of different approaches ranging from a straightforward Convolutional Neural Network over fine-tuned Large Language Models to few-shot-prompted language models using chain-of-thought reasoning. Surprisingly, we find that some individual ensemble members are not only more accurate than the final ensemble model but also more robust.

1 Introduction

In this paper, we describe our submission to SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials (NLI4CT 2024) (Jullien et al., 2024). In NLI4CT 2024, every model receives as input one or two clinical trial reports (CTRs) describing a breast cancer study. Further the model gets a hypothesis which makes a claim about the study and the section where the relevant information about the claim can be found in the CTR. Following a classical NLI setup (Bowman et al., 2015), the task of the model is to decide whether the hypothesis is logically entailed by the CTR or whether it contradicts the information in the CTR. NLI4CT 2024 is a continuation of a similar task that was held in 2023 (Jullien et al., 2023) and uses the same training and validation datasets. In contrast to the previous edition, NLI4CT 2024 focuses on the robustness of the submitted models. Specifically, it evaluates whether a model is consistent in its predictions and whether it predicts the correct label for the right reasons via targeted modifications of the test data; see Section 3 and Jullien et al. (2024) for more details.

We approach this task by building a large ensemble of diverse models. Our hypothesis is that ensembling a large variety of strong and weak models would improve robustness. For that we build ensembles of up to 25 models derived from 9 different approaches via different ensembling strategies. These approaches were implemented as part of a Master's course on biomedical Natural Language Processing at LMU Munich. Teams of two to three students chose a broad initial approach such as Convolutional Neural Networks (LeCun and Bengio, 1998; Kim, 2014) or data-centric machine learning (Swayamdipta et al., 2020). Then, they developed multiple models in the confines of the chosen approach while collaborating occasionally with other groups. Finally, evaluated all resulting models individually and as large ensembles on the test set. We find that ensembling generally improves robustness but that some individual approaches achieved even higher performance.

2 Methods

2.1 Approaches

We evaluate an ensemble of nine approaches. When selecting them, we favoured diversity over accuracy based on the assumption that even weaker models could contribute to the ensemble if they were diverse enough (Schapire, 1990). If not stated otherwise for a specific model, we used Adam (Kingma and Ba, 2015) for optimization. All approaches use only the section that contains the relevant information for inferring the NLI relation as provided by the task organizers.

Convolutional Neural Networks In the Convolutional Neural Networks (*CNN*, LeCun and Bengio (1998)) approach, we build on the work of Kim (2014). We modify this CNN-based model by replacing the word embeddings with subword embeddings from the embedding layer of

* Equal contribution. The order of the first-authors was chosen randomly.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1577–1583

BioBERT[1] (Lee et al., 2019). We train all models with Adam (Kingma and Ba, 2015), using a learning rate of $8.26e-6$ and maximum sequence length of 256. *CNN_1*: static cased BioBERT embeddings with kernels of size 3, 4, and 5 (100 each), a batch size of 32 and dropout of 0.5. *CNN_2*: static uncased BioBERT embeddings with kernels of size 3, 5, and 7 (100 each), a batch size of 32, dropout of 0.21 and weight decay of 0.001. *CNN_3*: static and dynamic cased BioBERT embeddings with kernels of size 3, 5, and 7 (100 each), a batch size of 64, dropout of 0.21 and weight decay of 0.001. *CNN_4*: static cased BioBERT embeddings, sequence length of 128, kernel sizes of 3 and 5 (100 each) and dropout of 0.21, trained for 10 epochs. *CNN_5*: static cased BioBERT embeddings, sequence length of 128, kernel sizes of 3, 4, 5 (50 each), batch size of 32, dropout of 0.21, trained for 20 epochs.

Fine-tuned transformers exploiting annotation biases With the *Bias* models, we attempt to exploit possible annotation biases following (Gururangan et al., 2018) who found that frequently a simple text classifier can decide the label for an instance based on the hypothesis alone. Specifically, we fine-tune a pre-trained language model to predict the NLI label using only the hypothesis as input. We optimze the hyperparameters with optuna using 10 runs per model. *Bias_1* uses BERT-base-cased[2] (Devlin et al., 2019) as model, *Bias_2* ClinicalBERT[3] (Wang et al., 2023), *Bias_3* BioBERT-PubMed200kRCT[4] (Deka et al., 2022), and *Bias_4* biomed_roberta_base[5] (Gururangan et al., 2020).

Diverse fine-tuned transformers For the Diverse fine-tuned transformers (*DT*) models, we fine-tune different pre-trained language models on the NLI4CT training data. After preliminary experiments with several transformer models, DeBERTa v3[6] (He et al., 2021) and BioLinkBERT[7] (Yasunaga et al., 2022) emerged as the most promising candidates. For both models, we used a maximum sequence length of 312, 20 epochs, and a learning rate of $2e^{-6}$. For *DT_1*, we use BioLinkBERT-base with a batch size of 4, for *DT_2*, BioLinkBERT-large with a batch size of 4, for *DT_3*, DeBERTa-v3-large with a batch size of 8, and for *DT_4*, DeBERTa-v3-base with a batch size of 4.

DeBERTa-v3 For the DeBERTa (*DeB_1*) model, we fine-tune DeBERTa-v3-large for 30 epochs, using a learning rate of 1e-5, a batch size of 8, and a max length of 312.

Stacking ensemble of two strong models For the *Ens* models, we construct an ensemble of two strong models. To construct this ensemble, we fine-tune DeBERTa-v3-large using a batch size of 8, a learning rate of 6e-6, a max length of 312, and 20 epochs. The other model in the ensemble is Mistral Instruct 7B v0.1[8] (Jiang et al., 2023), which we fine-tune on the NLI4CT training set to generate either "Entailment" or "Contradiction" using the prompt template proposed by Kanakarajan and Sankarasubbu (2023). We use a batch size of 8, a learning rate of 2e-4, and trained for 7.5 epochs. To enhance memory efficiency, we utilize the paged Adam optimizer, employ a sharded model and leverage QLoRa (Dettmers et al., 2023). To ensemble both models, we use both models to generate predictions on the development set of NLI4CT and then train a logistic regression classifier (James et al., 2013) to predict the correct label based on the predictions of both models. We experiment with providing additional metadata about the instance to the logistic regression classifier: the cosine distance between the TF-IDF representation of hypothesis and premise and the number of tokens in the concatenated hypothesis and premise. *Ens_1* is the full ensemble with metadata, *Ens_2* the ensemble without metadata, *Ens_3* only the Mistral model, and *Ens_4* only the DeBERTa model.

Data augmentation using hard instances In this approach, we follow Swayamdipta et al. (2020) and detect challenging data points using data maps with the goal of using this information for data augmentation. For this, we use a DeBERTa-v3 model that was pre-trained on various NLI datasets[9] (Lau-

[1] https://huggingface.co/dmis-lab/biobert-v1.1 / https://huggingface.co/dmis-lab/biobert-base-cased-v1.1

[2] https://huggingface.co/google-bert/bert-base-cased

[3] https://huggingface.co/medicalai/ClinicalBERT

[4] https://huggingface.co/pritamdeka/BioBert-PubMed200kRCT

[5] https://huggingface.co/allenai/biomed_roberta_base

[6] https://huggingface.co/microsoft/deberta-v3-base

[7] https://huggingface.co/michiyasunaga/BioLinkBERT-base

[8] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[9] https://huggingface.co/MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli

rer et al., 2022) and Flan-T5-base[10] (Chung et al., 2022). We fine-tune both models for 10 epochs on the shared task training data using a learning rate of 5e-5, a batch size of 16 and weight decay of 0.01 for DeBERTa and a learning rate of 2e-5, a batch size of 16, and weight decay of 0.001 for Flan. Then, we construct data maps from the resulting training dynamics and inspect hard-to-learn instances (low confidence and low variance) and ambiguous instances (medium-to-low confidence and high variance). We find that the models especially struggle with the following data characteristics: numerical reasoning, understanding synonyms (e.g. relating "cancer" and "carcinoma"), identifying hyponym/hyperonym relations (e.g. identifying "congestive heart failure" as a hypernym for "left ventricula systolic dysfunction), understanding abbreviations, and with specific sections in the CTR. We then use this information to manually construct 140 more instances that contain these specific issues which we use as additional training data. *Hard_1* is the DeBERTa-v3 model trained on the resulting dataset and *Hard_2* Flan-T5-base.

Data augmentation with GPTs for fine-tuned LLMs In this approach, we explore data augmentation with GPT-3.5 (Brown et al., 2020) and GPT-4 (OpenAI, 2023). We zero-shot prompt these models to generate 300 new statements and labels for randomly chosen CTRs. Then, we fine-tune a Mistral-Instruct-7B model on the training data augmented with these 300 new instances. For memory efficiency during fine-tuning, we employ QLora. We use a batch size of 50 and a learning rate of 2e-4. *Aug_1* is the model with the additional 300 new instances and *Aug_2* the same model fine tuned on the non-augmented data.

Fine-tuned LLMs with reasoning distillation from GPT-4 For the reasoning (*Reas*) models, we follow Wadhwa et al. (2023) and fine-tune a Mistral-7B model to use the reasoning of GPT-4 in order to generate the NLI label. For this, we 2-shot prompt GPT-4 to generate the label for all instances in the training data. We add the phrase *You should also show your reasoning process for your judgment* to the instruction and find that with this, GPT-4 generates texts that illustrate the steps involved in its reasoning process. Then, we filter out all 249 instances for which GPT-4 generated the wrong label and use the remaining 1451 as our new training set. Finally, we fine tune Mistral-7B for 8 epochs to generate the reasoning text together with the NLI label using a cosine-scheduled learning rate of 4e-4 and a batch size of 8. *Reas_1* is the model fine-tuned on the reasoning-augmented data whereas *Reas_2* is the same model fine-tuned on the original data.

Few-shot-prompted LLMs For the few-shot prompted LLM model (*Few_1*), we use Flan-T5-large[11] in a 1-shot prompting setting, where we show a randomly chosen example and ask it to generate the NLI label based on the CTR and the hypothesis.

2.2 Ensembling the approaches

We investigate six different variants to construct the ensemble which vary along two axes. The first axis is which models we include, because for most approaches we have multiple model variants. To construct our ensemble, we use a set of models $m \in \mathcal{M}$. For each model we have its predictions $\hat{y}_m \in \{-1, 1\}^n$ for all n test instances and its F1 score on the development set $F_1(m)$. We explore three heuristics to construct $\mathcal{M}$:

- Choose all available models *(all)*.
- For each approach, choose the model with the highest F1 score on the development set *(best)*.
- Choose the five models with the highest F1 score on the development set *(top-5)*. Note that multiple models can be based on the same approach.

The second axis is whether we use a simple majority vote or whether we weight models by their F1 score on the development set. Formally:

$$\hat{y} = sign[\sum_{m \in \mathcal{M}} \hat{y}\,] \quad \textit{(majority)} \quad (1)$$

$$\hat{y} = sign[\sum_{m \in \mathcal{M}} F_1(m) \cdot \hat{y}\,] \quad \textit{(weighted)} \quad (2)$$

We explore all possible combinations along these two axes leading to a total of six submitted ensemble models.

[10] https://huggingface.co/google/flan-t5-base

[11] https://huggingface.co/google/flan-t5-large

3 Evaluation protocol

NLI4CT 2024 uses three metrics to evaluate approaches. F1 score measured on the test set of NLI4CT 2023, consistency and faithfulness. Consistency measures whether the model always produces the same label for a set of instances that share the same meaning and thus the same gold label. Formally,

$$Consistency = \frac{1}{N'} \sum_{x'_i} 1 - \left| f(x_i) - f(x'_i) \right|, \tag{3}$$

where both x_i, x'_i share the same meaning and label and N' is the number of available x'_is. Faithfulness on the other hand scores whether the model is right for the right reasons. This metric considers correct predictions of the model and scores whether the model flips its prediction for instances in which semantic alterations lead to a flipped gold label:

$$Faithfulness = \frac{1}{\tilde{N}} \sum_{\tilde{x}_i} |f(x_i) - f(\tilde{x}_i)| \quad , \tag{4}$$

where the prediction for the original instance $f(x_i)$ is correct and $\tilde{x}_i$ is a semantic alteration of x_i that flips the gold label and $\tilde{N}$ is the number of available semantic alterations.

We evaluate all approaches on the hidden test set of NLI4CT 2024. We chose this approach even though frequent test set evaluation has severe downsides (van der Goot, 2021) because consistency and faithfulness could not be computed on the development set.

4 Results

Table 1 displays the results for all evaluated approaches. When considering the average of Test-F1, consistency, and faithfulness, the best performing model is *Reas_1* which fine-tunes Mistral-7b to following reasoning structures of GPT-4 before outputting the label. Its high average score is mainly due to a very high faithfulness score (85.8) paired with moderately high Test-F1 (76.0) and consistency (68.8) values. Its faithfulness is the 8th highest on the official leaderboard[12] whereas it ranks 13th/18th in terms of Test-F1/consistency. Notably, there is no clear winner across all metrics among the evaluated approaches. *Reas_2* achieves the best Dev-F1 score (82.0), *Ens_3* the best Test-F1 (76.8), and *Ens_4* the best consistency (72.0).

[12] https://codalab.lisn.upsaclay.fr/competitions/16190#results

name	Dev	Test	Cons	Faith	Avg
CNN_1	60.0	47.7	57.7	63.0	56.1
CNN_2	56.0	55.5	51.4	39.4	48.7
CNN_3	61.0	49.2	55.9	57.2	54.1
CNN_4	52.0	53.0	54.1	53.2	53.5
CNN_5	58.0	43.5	57.2	71.5	57.4
Bias_1	63.0	45.1	58.8	71.9	58.6
Bias_2	58.0	54.3	51.6	45.6	50.5
Bias_3	61.0	48.2	57.6	65.3	57.0
Bias_4	66.0	53.5	57.2	56.1	55.6
DT_1	67.0	55.7	59.8	63.5	59.7
DT_2	67.0	55.4	51.2	40.7	49.1
DT_3	76.0	71.9	64.8	66.2	67.6
DT_4	76.0	71.9	64.8	66.2	67.6
DeB_1	77.0	72.4	64.7	54.1	63.7
Ens_1	76.0	74.1	70.1	72.9	72.4
Ens_2	76.0	73.2	71.0	83.3	75.9
Ens_3	76.0	**76.8**	67.4	65.2	69.8
Ens_4	78.0	73.4	**72.0**	74.0	73.1
Hard_1	72.0	18.1	48.3	71.6	46.0
Hard_2	59.0	61.5	54.7	49.0	55.1
Aug_1	69.0	64.6	62.5	71.2	66.1
Aug_2	74.0	68.8	64.7	75.9	69.8
Reas_1	76.0	74.7	68.8	85.8	**76.4**
Reas_2	**82.0**	75.9	67.1	76.7	73.3
Few_1	42.0	28.6	60.7	**86.5**	58.6
all_maj	-	70.1	68.6	76.0	71.6
all_wei	-	72.3	**69.9**	73.1	71.8
best_maj	-	70.4	69.4	**81.6**	**73.8**
best_wei	-	**74.2**	70.3	72.8	72.4
top5_maj	-	70.1	68.6	76.0	71.6
top5_wei	-	72.3	**69.9**	73.1	71.8

Table 1: NLI4CT 2024 test set results for all our evaluated approaches in percent. *Cons.* is consistency, *Faith.* is faithfulness and *Avg.* is the average over all three. Individual approaches are the top part of the table whereas the six diverse ensemble approaches are at the bottom. Models included in *best* ensembles are in bold and models included in *top5* are additionally in italics. The highest score per column is in bold.

Large ensemble results Interestingly, *Reas_1* achieves an even better average score (76.4) than the best large ensemble model *best_majority* (73.8). Generally, in terms of average performance, the best large ensemble outperforms all but two single approaches, *Ens_2* and *Reas_1*, where *Ens_2* itself is an ensemble of two strong models and *Reas_1* combines two models via distillation. Furthermore, neither *Reas_1* nor *Ens_2* were included in the *best_majority* ensemble because their Dev-F1 scores were lower than those of other models from the same approach. Based on these observations, we can confirm our initial hypothesis that building a large ensemble improves the average performance. However, for consistency and faithfulness other individual approaches perform better than the large ensembles. In terms of average scores, taking the *best* model per approach performs clearly better than taking *all* or only the *top*5.

Dev-F1 as model selection criterion Unsurprisingly, using only Dev-F1 as the criterion for model selection and hyperparameter tuning is not sufficient for maximizing the average performance over Test-F1 consistency, and faithfulness. In five out of nine approaches, the model that achieves the best Dev-F1 score does not achieve the best average score. This also has consequences for our best-performing ensembling approach *best* because it implies that in five out of nine cases we include a suboptimal model in our ensemble. This suggests that using a development set that allows for measuring consistency and faithfulness for model selection, hyperparameter tuning or ensemble construction could improve these properties at test time.

Overlap between approaches We analyze how similar the predicions of different approaches are. For that, we compute the pairwise Cohen's kappa scores between all evaluated models. A heatmap of the results can be found in Figure 1. As expected, models stemming from the same approach produce similar results, as can be seen from the bright squares around the diagonal of the heatmap. Additionally, the predictions of the CNN models correlate with those of the Bias models, suggesting that the *CNNs* might also mainly consider the hypothesis and disregard context information. Another notable group of correlations is that between the large ensemble and some of the *DT*, the *DeB*, the *Ens*, and the *Reas* models. This could indicate that most of the large ensemble models mainly rely on the predictions of these strongly performing models.

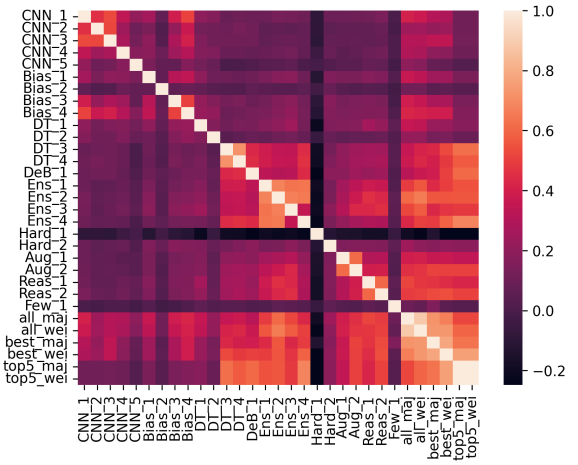


Figure 1: Pairwise Cohen's kappa scores between all evaluated methods.

5 Conclusion

This paper describes our contribution to the SemEval-2024 NLI4CT shared task on robust NLI for clinical trial reports. We investigate whether a large diverse ensemble can improve robustness. Our results largely confirm this hypothesis, but we find that some individual approaches perform even better and more robust than our best ensemble.

In this work, we investigated only ensembling based on voting procedures and completely disregarded the confidences of the individual models. Further, we did not use more sophisticated approaches such as stacking. Finally, we used data-centric approaches only to augment the training data of individual models, but did not use it to evaluate the robustness of models. We believe that all of these could potenetially improve the accuracy and robustnes of NLI models for clinical trial reports.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child,

Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Pritam Deka, Anna Jurek-Loughrey, et al. 2022. Evidence extraction to validate medical claims in fake news detection. In *International Conference on Health Information Science*, pages 3–15. Springer.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. 2013. *An introduction to statistical learning*, volume 112. Springer.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama AI research at SemEval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003, Toronto, Canada. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Welbers Kasper. 2022. Less annotating, more classifying – addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert - nli. *preprint*.

Yann LeCun and Yoshua Bengio. 1998. *Convolutional networks for images, speech, and time series*, page 255–258. MIT Press, Cambridge, MA, USA.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Robert E. Schapire. 1990. The strength of weak learnability. *Mach. Learn.*, 5:197–227.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Rob van der Goot. 2021. We need to talk about train-dev-test splits. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4485–4494, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Guangyu Wang, Xiaohong Liu, Zhen Ying, Guoxing Yang, Zhiwei Chen, Zhiwen Liu, Min Zhang, Hongmei Yan, Yuxing Lu, Yuanxu Gao, et al. 2023. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. *Nature Medicine*, 29(10):2633–2642.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

MARiA at SemEval 2024 Task-6: Hallucination Detection Through LLMs, MNLI, and Cosine similarity

Reza Sanayei* **Abhyuday Singh*** **MohammadHossein Rezaei** **Steven Bethard**
University of Arizona
rsanayei,abhyudaysingh,mhrezaei,bethard@arizona.edu

Abstract

The advent of large language models (LLMs) has revolutionized Natural Language Generation (NLG), offering unmatched text generation capabilities. However, this progress introduces significant challenges, notably hallucinations—semantically incorrect yet fluent outputs. This phenomenon undermines content reliability, as traditional detection systems focus more on fluency than accuracy, posing a risk of misinformation spread.

Our study addresses these issues by proposing a unified strategy for detecting hallucinations in neural model-generated text, focusing on the SHROOM task in SemEval 2024. We employ diverse methodologies to identify output divergence from the source content. We utilized Sentence Transformers to measure cosine similarity between source-hypothesis and source-target embeddings, experimented with omitting source content in the cosine similarity computations, and Leveragied LLMs' In-Context Learning with detailed task prompts as our methodologies. The varying performance of our different approaches across the subtasks underscores the complexity of Natural Language Understanding tasks, highlighting the importance of addressing the nuances of semantic correctness in the era of advanced language models.

1 Introduction

The SHROOM task (Mickus et al., 2024) aims to address the challenge of detecting grammatically sound outputs containing incorrect semantic information in NLG systems. This task is crucial due to the prevalent issue of neural models producing fluent but inaccurate outputs, referred to as "hallucinations" (Maynez et al., 2020). Given the critical importance of correctness in NLG applications, SHROOM aims to foster interest in automating the detection of these hallucinations. Participants were tasked with the binary identification of such hallucinations across different NLG tasks, including Definition Modeling (DM), Machine Translation (MT), and Paraphrase Generation (PG).

Our system leveraged three distinct approaches to tackle the SHROOM task: A baseline cosine similarity, MultiNLI classification (Williams et al., 2018), and Large Language Models (LLMs), specifically Mixtral-8x7B-Instruct (Jiang et al., 2024). Each approach was tailored to identify hallucinations in NLG outputs by comparing them with the source input and detecting inconsistencies in semantic information. Through various combinations of these approaches, we aimed to accurately identify grammatically sound but incorrect outputs generated by neural models.

2 Background and Related Work

Previous research efforts have attempted to detect and control (Filippova, 2020) hallucinations. Dziri et al. (2022) worked on the origins of hallucinations, concluding that $> 60\%$ of the standard benchmarks consist hallucinated responses. Xiao and Wang (2021) proposed a simple extension to beam search to reduce hallucination. Obaid ul Islam et al. (2023) proposed a natural language inference (NLI) based method to preprocess the training data to reduce hallucinations.

The most similar to our work, Guerreiro et al. (2023) studied hallucinations in Neural Machine Translation. They analyzed multiple methods to detect hallucinations and developed `DeHallucinator` which overwrites the translation detected as a hallucination with a better one.

3 Dataset

In this section, we provide a detailed overview of the SHROOM dataset, highlighting its composition, structure, and associated challenges.

* Equal Contribution.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1584–1588

3.1 Composition

The SHROOM dataset consists of two main tracks: model-aware and model-agnostic, encompassing three subtasks: paraphrase generation, machine translation, and definition modeling. The test dataset comprises 3000 objects, with 1500 belonging to each of the model-aware and model-agnostic tracks. Additionally, the development data consists of 500 objects for each track, while the trial data comprises 80 objects. The unlabeled training data comprises 5000 objects for both the model-aware and model-agnostic tracks.

3.2 JSON Object Structure

Each dataset object contains the following components:

Task Description: Indicates the subtask to which the object belongs. Source (src): Input passed to be processed by the NLP model. Target (tgt): Intended correct processed "gold" text. Hypothesis (hyp): Actual output produced by the NLP model. Reference (ref): Specifies whether the reference includes the source, target, both, or neither. Labels: Each object is labeled by five human annotators as hallucination or not hallucination. Probability of Hallucination (p(hallucination)): Represents the probability of the hypothesis being a hallucination, ranging from 0.0 to 1.0. This probability is determined based on the consensus of the annotators. Label: Indicates the majority vote among the annotators, labeling the object as hallucination or not hallucination.

3.3 Issues with the Dataset

Several challenges were encountered while working with the SHROOM dataset:

Unlabeled Training Data: The unlabeled nature of the training data posed challenges, limiting the applicability of certain approaches and requiring alternative strategies for model training. **Format Discrepancy in Definition Modeling Task:** The test data for the definition modeling task deviated from the format of the development data, missing the <define> tag and presenting the definition as a question at the end. This inconsistency caused issues in several approaches and led to hallucinations in the Large Language Model (LLM) approach. **Imbalance in Language Representation:** The machine translation subtask lacked a balanced representation of multiple languages, potentially skewing the evaluation results and posing challenges for system development.

4 System Overview

In this section, we provide an overview of the three approaches employed in our system to address the SHROOM task.

4.1 Baseline Approach - Cosine Similarity

Our baseline approach utilizes cosine similarity to compare embeddings derived from source-hypothesis and source-target pairs. We employ Sentence Transformers (Reimers and Gurevych, 2019) to compute the cosine similarity, facilitating the comparison between the generated hypothesis and the target output. This approach serves as the foundation upon which subsequent refinements are built.

4.2 Approach 2 - MNLI Classification

In this approach, we leveraged the MultiNLI (MNLI) dataset for classification and similarity comparison between hypothesis and target outputs. We utilized the bart-large-mnli model, which is pre-trained on MNLI, to predict the entailment relationship between the hypothesis and target, subsequently examining similarity, and predict hallucination.

4.3 Approach 3 - Large Language Models

Leveraging LLMs, we prompt-engineered instructions for each subtask, utilizing the In-Context Learning power of these models, specifically Mixtral-8x7B-Instruct model, to detect hallucinations. We use the model for inference and experiment with temperature adjustments to optimize performance. This approach capitalizes on the contextual understanding and generative capabilities of LLMs to accurately identify hallucinations in NLG outputs.

5 Experimental Setup

In this section, we outline the experimental setup used for evaluating our system's performance on the SHROOM task.

5.1 Data Splits

We utilized the provided development set extensively for experimentation, as the training set was unlabeled. This allowed us to iteratively refine our approaches before selecting the final submissions for the test set evaluation.

Task	Hypothesis	Reference	Source	Target	Model	Labels	Label	P(H)
DM	(linguistics) The study of the relationships between words and their meanings.	Target	The <define> metaontology </define> debate has now migrated from discussions of composition.	The ontology of ontology.	-	H/N/N/N	N	0.4
PG	When did you see him?	Either	When did you last see him?	When was the last time you saw him?	tuner007/ pegasus_paraphrase	N/N/N	N	0.0
MT	It uses a giant rocket over 100 feet high to launch a satellite or telescope into space.	-	Ngini mamakai roket raksasa malabihi 100 kaki tingginya gasan maandak satelit atawa teleskop ka luar angkasa.	It takes a giant rocket over a 100 feet high to put a satellite or telescope in space.	-	N/N/N/H/N	N	0.4

Table 1: Examples from the dataset. The dataset includes three subtasks: Definition Modeling (DM), Paraphrase Generation (PG), and Machine Translation (MT). The dataset is labeled by crowdworkers as Hallucination (H) and Not Hallucination (N). P(Hallucination) indicates the probability of the hallucination based on the labels.

5.2 Preprocessing and Model Selection

Minimal preprocessing was applied to the data. Furthermore, we did not create separate distinctions for the model-aware and model-agnostic subtasks. Our decision was driven by the belief that a unified, model-agnostic solution would be the most optimal approach for addressing the task. For cosine similarity, we employed Sentence Transformers. MNLI classification utilized the Facebook bart-large-mnli model. In the case of LLMs, we initially employed Mixtral-8x7B-Instruct model and conducted an additional run post-evaluation with some changes to model settings like output token size and temperature, processing queries in batches.

5.3 Evaluation Measures

The evaluation measures used in the task primarily revolved around accuracy percentages. We assessed the accuracy of our models in correctly identifying grammatically sound outputs containing incorrect or unsupported semantic information, inconsistent with the source input. This metric served as the primary indicator of our system's performance on the SHROOM task.

This Experimental Setup section provides essential details about our methodology and the specifics of our experimental setup, enabling reproducibility and facilitating a clear understanding of our system's performance on the SHROOM task.

6 Results

In this section, we present the quantitative analysis of our system's performance on the SHROOM task. We evaluated the performance of our system approaches on the test data for each of the three subtasks: Paraphrase Generation (PG), Machine Translation (MT), and Definition Modeling (DM). Our system comprises three distinct approaches: cosine similarity, MNLI classification, and Large Language Models (LLMs), specifically Mixtral. **We note that Mixtral is the only system submitted to the task, and other results are post-evaluation experiments.**

6.1 Model-Agnostic Setting

Table 2 provides the accuracy of model-agnostic setting. We observe that our cosine similarity approach achieved the highest accuracy, with 70.3% overall accuracy. Specifically, it performed well in PG (77.9%) and MT (75.8%) subtasks. However, the Mixtral approach yielded lower accuracy at 50.5%, with varying performance across subtasks: DM (48.4%), PG (50.1%), and MT (52.9%). After changing the settings (temperature) the accuracy improved to 60.2% (Mixtral*).

6.2 Model-Aware Setting

Table 3 provides the accuracy of model-aware setting. In the model-aware setting, the MNLI classification approach achieved the highest accuracy at

	Cosine	MNLI	Mixtral	Mixtral*
Accuracy	62.1	65.13	49.8	56.1
DM	60.4		48.3	53.5
PG	57.6		48.2	56.1
MT	66.7		52.3	58.7

Table 2: Results on model-agnostic setting. We report accuracy for all the instances and accuracy on each subtask.

	Cosine	Mixtral	Mixtral*
Accuracy	70.3	50.5	60.2
DM	59.8	48.4	56.8
PG	77.9	50.1	61.9
MT	75.8	52.9	62.2

Table 3: Results on model-aware setting. We report accuracy for all the instances and accuracy on each subtask.

65.13%, followed by the cosine similarity approach at 62.1%. The MNLI approach showed consistent performance across subtasks, while the cosine similarity approach performed particularly well in MT (66.7%). The Mixtral approach had the lowest accuracy at 49.8%, with varying performance across subtasks: DM (53.5%), PG (56.1%), and MT (58.7%). After changing the settings— (temperature) the accuracy improved to 56.1% (Mixtral*).

7 Conclusion

In conclusion, our system for addressing the SHROOM task employed three distinct approaches: baseline cosine similarity, MNLI classification, and Mixtral. Each approach was carefully designed to tackle the challenge of identifying hallucinations in natural language generation outputs.

Our experimental results demonstrated varying degrees of success across the different subtasks. While cosine similarity and MNLI classification showed promising performance, leveraging LLMs proved to be particularly effective in accurately identifying hallucinations.

Looking forward, our system's performance suggests several avenues for future work. Firstly, further exploration and refinement of each approach tailored to the specific subtleties of each subtask could potentially yield improved performance. Additionally, investigating ensemble methods or hybrid approaches that combine the strengths of different techniques may enhance overall system robustness.

Despite the challenges encountered, our system's competitive performance in the SHROOM task underscores the importance of automated, multi-expert, mechanisms for detecting and mitigating hallucinations in NLG systems. As the field continues to evolve, addressing these challenges will be crucial for advancing the reliability and accuracy of NLG applications.

In summary, our system represents a significant step towards addressing the complexities of hallucination detection in NLG outputs, and we are optimistic about the potential for future advancements in this area.

References

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.

Katja Filippova. 2020. Controlled hallucinations: Learning to generate faithfully from noisy data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 864–870, Online. Association for Computational Linguistics.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024.

SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Saad Obaid ul Islam, Iza Škrjanec, Ondrej Dusek, and Vera Demberg. 2023. Tackling hallucinations in neural chart summarization. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 414–423, Prague, Czechia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

NUS-Emo at SemEval-2024 Task 3: Instruction-Tuning LLM for Multimodal Emotion-Cause Analysis in Conversations

Meng Luo[1*] **Han Zhang**[2*] **Shengqiong Wu**[1] **Bobo Li**[3] **Hong Han**[2] **Hao Fei**[1†]
[1]National University of Singapore [2]Xidian University [3]Wuhan University
mluo@u.nus.edu zhanghanxd@stu.xidian.edu.cn swu@u.nus.edu
boboli@whu.edu.cn hanh@mail.xidian.edu.cn haofei37@nus.edu.sg

Abstract

This paper describes the architecture of our system developed for Task 3 of SemEval-2024: Multimodal Emotion-Cause Analysis in Conversations. Our project targets the challenges of subtask 2, dedicated to Multimodal Emotion-Cause Pair Extraction with Emotion Category (MECPE-Cat), and constructs a dual-component system tailored to the unique challenges of this task. We divide the task into two subtasks: emotion recognition in conversation (ERC) and emotion-cause pair extraction (ECPE). To address these subtasks, we capitalize on the abilities of Large Language Models (LLMs), which have consistently demonstrated state-of-the-art performance across various natural language processing tasks and domains. Most importantly, we design an approach of emotion-cause-aware instruction-tuning for LLMs, to enhance the perception of the emotions with their corresponding causal rationales. Our method enables us to adeptly navigate the complexities of MECPE-Cat, achieving a weighted average 34.71% F1 score of the task, and securing the $\mathbf{2}^{nd}$ rank on the leaderboard.[1] The code and metadata to reproduce our experiments are all made publicly available.[2]

1 Introduction

Emotion cause analysis is a critical component of human communication and decision-making, offering substantial applications across diverse fields. It enables a deeper and more detailed understanding of sentiments. The introduction of emotion-cause analysis in textual conversations by Poria et al. (2021); Xia and Ding (2019) has paved the way for advancements in understanding emotional dynamics within dialogues. However, textual analysis alone does not fully capture the complexity of human emotional expression, as emotions and their causes are often conveyed through a blend of modalities (Hazarika et al., 2018; Wu et al., 2023a; Fei et al., 2023b). Subtask 2 of SemEval-2024 Task 3, referred to as MECPE-Cat, seeks to expand this analysis into the multimodal domain, focusing on English-language conversations. The task draws inspiration from the seminal work of Wang et al. (2023), which sets out to jointly extract emotions and their corresponding causes from conversations across multiple modalities, including text, audio, and video, and it also encompasses the identification of the corresponding emotion category for each emotion-cause pair.

In our system, we leverage LLMs such as GPT-3 (Brown et al., 2020), Flan-T5 (Chung et al., 2022), and GLM (Du et al., 2021) known for their exceptional performance in various natural language processing tasks. We employ parameter-efficient fine-tuning, specifically LoRA (Hu et al., 2021), to efficiently fine-tune LLMs, enhancing their performance with minimal computational overhead. Additionally, we harness emotion-cause-aware prompt-based learning and instruction-tuning to enhance model performance such that the LLMs can more accurately perceive the emotions with their corresponding causal rationales. Prompt-based learning guides LLMs to generate contextually relevant outputs, while instruction-fine-tuning models for our specific tasks by improving their response to explicit instructions.

In this paper, we investigate the optimal LLM for the MECPE-Cat task, selecting ChatGLM based on its superior zero-shot performance. We further refine ChatGLM through instruction-tuning, using carefully crafted prompts to enhance its task-specific accuracy. Our fine-tuned model achieves the second-highest score on the official test set for subtask 2, with a weighted average of 34.71% F1,

[*]Equal contributions.
[†]Corresponding author.
[1]https://nustm.github.io/SemEval-2024_ECAC/
[2]https://github.com/zhanghanXD/NUS-Emo-at-SemEval-2024-Task3

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1589–1596
June 20-21, 2024

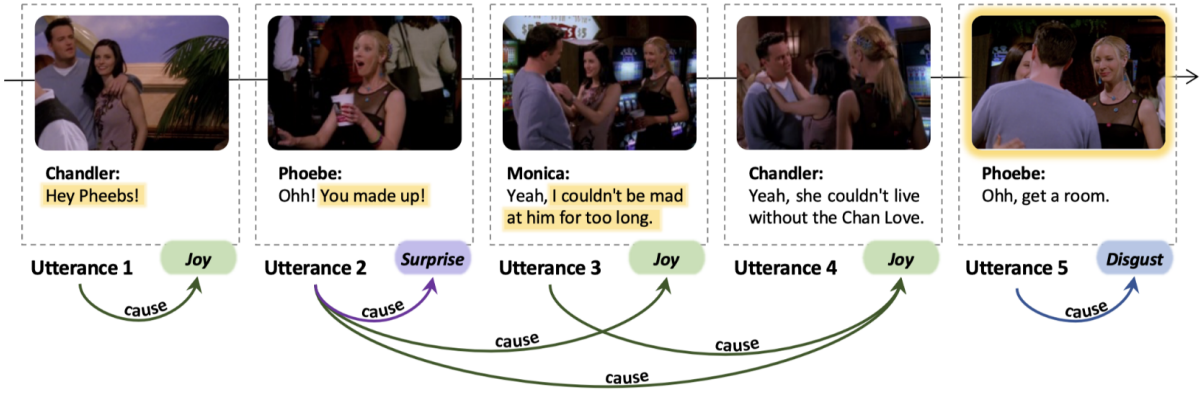


Figure 1: An example of an official task and annotated dataset. Each arc points from the cause utterance to the emotional triggers. The cause spans have been highlighted in yellow. Background: Chandler and his girlfriend Monica walked into the casino (they had a quarrel earlier but made up soon), and then started a conversation with Phoebe.

underscoring the effectiveness of our approach. We also discuss the current limitations of our model and methodology, alongside directions for future research and improvement. We will release our codes and resources mentioned in this paper to facilitate relevant research.

2 Background

2.1 Task and Dataset Description

The SemEval-2024 Task 3 (Wang et al., 2024) is based on the multimodal conversational emotion-cause dataset, Emotion-Cause-in-Friends (ECF; Wang et al., 2023), by choosing a multimodal dataset MELD (Poria et al., 2018) as the data source and further annotating the corresponding causes for the given emotion annotations. The ECF dataset contains 9,794 emotion-cause pairs, covering three modalities. The subtask 2 is to extract all emotion-cause pairs in a given conversation under three modalities, where each pair contains an emotion utterance along with its emotion category and a cause utterance, e.g., (U3_Joy, U2), which means that the speaker's joy emotion in utterance 3 is triggered by the cause from utterance 2. Figure 1 displays a real example of this task and annotated dataset. In this conversation, it is expected to extract a set of six utterance-level emotion-cause pairs in total, e.g., Chandler's Joy emotion in Utterance 4 (U4 for short) is triggered by the objective cause that he and Monica had made up and Monica's subjective opinion in U3, forming the pairs (U4_joy, U2) and (U4_joy, U3); The cause for Phoebe's Disgust in U5 is the objective event that Monica and Chandler were kissing in front of her (mainly reflected in the visual modality of U5), forming the pair (U5_disgust, U5).

2.2 Related Work

The exploration of ECPE within textual and conversational contexts has been approached through various methodologies, each tailored to specific task settings (Chen et al., 2022). Cheng et al. (2023) reframe the ECPE task as a process akin to engaging in a two-stage machine reading comprehension (MRC) challenge. Zheng et al. (2023) expand the ECPE task to Emotion-Cause Quadruple Extraction in Dialogs (ECQED), focusing on detecting pairs of emotion-cause utterances and their types. They present a model utilizing a heterogeneous graph and a parallel grid tagging scheme for this purpose. In addressing the specific challenge of the MECPE-Cat task, Wang et al. (2023) set a benchmark for this task by introducing two preliminary baseline systems. They utilize a heuristic approach to leverage inherent patterns in the localization of causes and emotions, alongside a deep learning strategy, MECPE-2steps, which adapts a prominent ECPE methodology for news articles to include multimodal data.

Drawing from the varied methodologies of previous work, it becomes clear that effectively solving the MECPE-Cat task demands a deep understanding of dialogue content, precise identification of conversational emotions, extraction of emotion-cause pairs, and the integration of multimodal information. Motivated by the strong performance of LLMs on various metrics, we opt to utilize these models to address this intricate challenge. Through exhaustive model evaluations and extensive prompt

testing, we have showcased the practicality, superiority, and adaptability of our chosen approach.

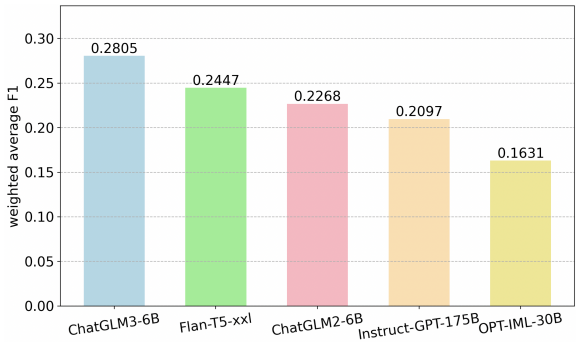


Figure 2: Zero-shot test set performance of various instruction-tuned LLMs.

3 Methodology

In this section, we first conduct preliminary experiments to determine which LLM to select as a backbone reasoner. We then elaborate on how we design the system and emotion-cause-aware instructions for tuning our chosen LLM.

3.1 Pilot Study for LLM Selection

Currently, there exists a variety of LLMs, such as OPT-IML (Iyer et al., 2022), GPT-3, Flan-T5, and GLM. However, it is essential to select a model that not only performs optimally but is also the most suitable for our specific task. To this end, we carry out a pilot study to determine the most appropriate model selection. For our zero-shot testing experiment, we rigorously evaluate several models, including OPT-IML[3], Instruct-GPT[4] (Ouyang et al., 2022), Flan-T5[5], alongside the ChatGLM models, to identify the most effective tool for this specific task. We customize instructions for each model's specific tuning style, recognizing that a single set of instructions does not suit all models effectively. We also embed expected output labels within these instructions to secure precise responses from each model. Figure 2 depicts the zero-shot performance of these models. The ChatGLM[6] LLM is ultimately selected based on its superior performance in these tests. This selection is informed not merely by the innovative features or the advanced training methodologies of ChatGLM but by empirical evidence of its exceptional zero-shot performance among the models considered.

[3]OPT-IML-30B, max version with 30B, https://huggingface.co/facebook/opt-iml-30b

[4]Instruct-GPT-175B, an advanced version of the GPT-3.5.

[5]Flan-T5-xxl, with 11B, https://huggingface.co/google/flan-t5-xxl

[6]ChatGLM, 3rd version with 6B, https://github.com/THUDM/ChatGLM3.

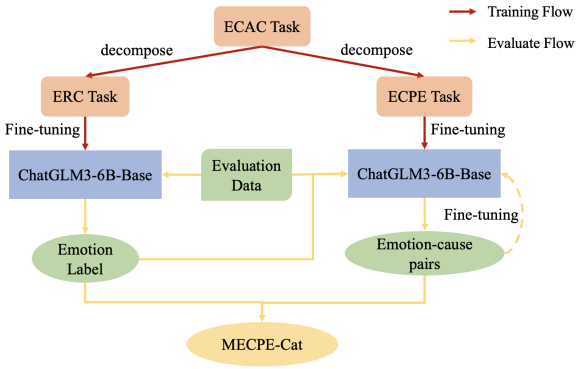


Figure 3: Proposed method workflow for the MECPE-Cat task.

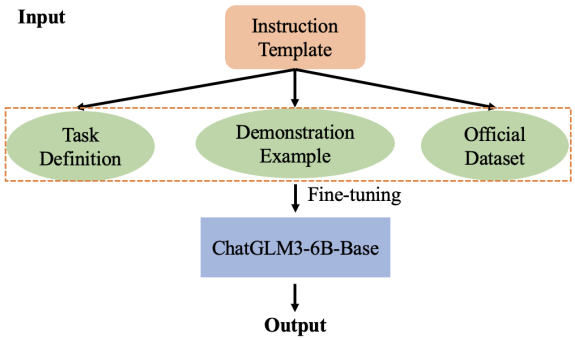


Figure 4: The construction of the instruction template and the flow of model input and output.

3.2 Multimodal Feature Encoding

Given that the inputs for our task incorporate multimodal signals, including visual information to assist in more accurate emotion recognition, it is imperative to fully leverage the non-textual modal information. However, our LLM backbone does not natively support the direct inclusion of non-textual modal signals. To address this, we consider employing ImageBind (Girdhar et al., 2023) for encoding the multimodal portion of input information, owing to its robust multimodal alignment capabilities and visual perception proficiency. Subsequently, we concatenate the multimodal representations with other textual embeddings before feeding them into the LLM.

3.3 Constructing Emotion-Cause-aware Instructions for LLM Tuning

Figure 3 first illustrates the workflow of our proposed framework. Initially, we fine-tune the model on the ERC task. Following this, we incorporate

the predicted emotion labels into each utterance, setting the stage for the ECPE task execution. Subsequently, we employ the model, now fine-tuned with data labeled with emotion tags, to perform inference on the MECPE-Cat task, yielding an initial set of emotion-cause pairs. These preliminary results are then reintegrated into the original training dataset for a second round of fine-tuning, culminating in the refinement of our model to produce the final set of emotion-cause pairs.

Task Definition:
"You're an expert in sentiment analysis and emotion cause identification. Below is a conversation containing multiple utterances from different speakers, along with the corresponding emotion label for each utterance. Your task is to identify the indices of the candidate utterances that elicited the emotion in the target utterance."

Input conversation:
1_joy. Chandler: Hey Pheebs!
2_surprise. Phoebe: Ohh! You made up!
3_joy. Monica: Yeah, I couldn't be mad at him for too long.
4_joy. Chandler: Yeah, she couldn't live without the Chan Love.
5_disgust. Phoebe: Ohh, get a room.

Candidate utterances:
1_joy. Chandler: Hey Pheebs!
2_surprise. Phoebe: Ohh! You made up!
3_joy. Monica: Yeah, I couldn't be mad at him for too long.

Target utterance:
4_joy. Chandler: Yeah, she couldn't live without the Chan Love.

Question:
The emotion-cause indices of the target utterance are:
[LLM output]

To enhance the perception of identifying emotion-cause pairs and mitigate the task's inherent complexity and potential confusion, we design the template for producing emotion-cause-aware instructions to guide the model. Figure 4 illustrates the construction of the instruction template, which encompasses the task definition, a demonstration example, and the dataset for which the model is expected to predict outcomes. This structured approach not only simplifies the task's complexity for the model but also aligns the model's processing capabilities with the requirements of accurately identifying emotion-cause pairs in conversations. In the above box we showcase a real example.

4 Experiments

This section will quantify the effectiveness of our systems via experiments and also show more analyses to gain more observations.

4.1 Implementation

The hyperparameter of our system used to achieve the highest weighted average F1 score on the subtask 2 is listed in 1. The ChatGLM model was fine-tuned using a learning rate of 1e-4 with LoRA-specific configurations including a rank of 8, alpha value of 32, and a dropout rate of 0.1. The training was conducted with a maximum instruction length of 2048 tokens and an output length limited to 128 tokens, using a batch size of 1. We used a single gradient accumulation step across 2 training epochs. These parameters were meticulously selected to optimize our model's performance.

Hyperparameter	Value
Learning rate	1e-4
LoRA rank	8
LoRA alpha	32
LoRA dropout	0.1
Max instruction length	2048
Max output length	128
Batch size	1
Gradient accumulation steps	1
Epochs	2

Table 1: Hyperparameter used for the best performing model.

4.2 Evaluating Template Designing

In constructing the instruction dataset for tuning LLMs, we systematically transform each dialogue in the dataset into training samples by embedding them into a fixed template as described above. The data source for this transformation is the officially provided ECF dataset, which comprises 13,619 utterances. Consequently, we constructed a total of 13,619 templates based on this dataset, each

Condition	F1 Score
Only Task Definition	0.2981
Task + Example	0.3124
Task + Example + Candidate	**0.3207**

Table 2: Performance using different templates for constructing instruction tuning.

tailored to facilitate the model's learning and application of emotion-cause-aware instructions.

We here perform an ablation study on the contributions of each part of the instructions we designed for the task. We derive three variants:

- **Only Task Definition:** Compared to the zero-shot paradigm, this condition offers a more detailed and precise description of the task.
- **Task + Example:** We provide a demonstrative example to clearly show the expected outcome in a real-world dialogue, offering the model a practical reference for task execution
- **Task + Example + Candidate utterances:** This design simplifies the task by introducing 'candidate utterances,' enabling the model to analyze emotion-cause pairs sentence by sentence, rather than across entire dialogues, and pinpoint the specific causes of emotions from the preceding content.

Table 2 demonstrates the comparative performance of these diverse templates. We see that different components of the instruction templates show clear influences, such as task definition, example demonstration, and candidate utterances. Thus, we apply all these components into our instruction templates.

4.3 Instruction-tuning LLM

For our experiments, we adopt a meticulous fine-tuning process for the ChatGLM. We set a learning rate of 1e-4, aiming for a balance between rapid convergence and maintaining the model's ability to adapt without overfitting. We leverage the LoRA technique with a rank of 8 and alpha of 32 to introduce task-adaptive parameters without bloating the model size, alongside a dropout rate of 0.1 to prevent overfitting. The model processed inputs with a max sequence length of 2048 tokens, accommodating the depth of context required for our task, while the outputs are capped at 128 tokens to focus on generating concise and relevant responses. Both batch size and gradient accumulation steps are set to 1, tailored to our computational resources while ensuring effective backpropagation. This configuration, selected after careful evaluation of various setups, is instrumental in fine-tuning the ChatGLM model to achieve the best performance on our task.

Our experiments capitalize on the robust computational capabilities provided by NVIDIA A800-SXM GPUs, each boasting 80 GB of VRAM, to ensure sufficient resources are available to train large language models. This fine-tuning process is facilitated using a customized script derived from the Hugging Face Transformers framework, chosen for its extensive support of transformer models and seamless integration with our setup, thereby enabling us to leverage advanced hardware capabilities while utilizing a leading-edge software environment for our model's optimization.

4.4 Task Decomposition

We decompose the MECPE-Cat task into ERC and ECPE phases to strategically alleviate its complexity. This division offered a two-fold advantage: firstly, it distills the task into clearer, more focused components, facilitating a more straightforward understanding and execution of the model. Secondly, by leveraging emotion labels obtained from the ERC phase during the ECPE phase, we enhance the model's capability to pinpoint emotion-cause pairs with greater accuracy. Tabel 3 showcases incremental improvements in weighted average F1 scores across three distinct setups. This progression underscores the dual benefits of our approach: simplifying the task's complexity for the model and enriching the ECPE phase with contextual emotion labels, thereby optimizing the extraction of emotion-cause pairs.

Methods	F1 Score
Single Stage	0.3207
Two Independent Stages	0.3288
ECPE with Emotion Labels	**0.3396**

Table 3: Comparison of weighted average F1 Scores under different methods.

4.5 Data Augmentation

We find that augmenting the training dataset with trial data significantly enhanced model accuracy, achieving a high weighted average F1 score of 0.3416, as shown in Table 4. Furthermore, we employ a trick by incorporating the model's inference results on the ECPE task back into the training dataset for an additional round of fine-tuning. This iterative fine-tuning strategy yielded a further improvement in our test data performance. These

enhancements demonstrate the efficacy of not only expanding the training dataset but also utilizing the model's own outputs to refine its accuracy.

Data	Epoch 1	Epoch 2	Epoch 3
Train	0.3390	0.3396	0.3393
Train + Trial	0.3404	0.3410	0.3406
Iterative Train	0.3408	**0.3416**	0.3411

Table 4: Comparison of weighted average F1 Scores across different training data and epochs.

4.6 Multimodal Integration

To assess the impact of multimodal information on our model's performance, we adopt a methodological approach that harnessed GPT-4V Achiam et al. (2023) for extracting insights from modalities beyond text. Specifically, we enrich the instruction template with "video description of target utterance" derived from GPT-4V, presenting it as supplementary information to guide the model. This strategic integration of multimodal data leads to an improvement in the model's F1 score, as shown in Table 5, which validates the utility of multimodal information in providing richer contextual understanding.

Information	F1 Score
Text	0.3416
Text + Video	**0.3471**

Table 5: Comparison of weighted average F1 Scores between pure text and multimodal information.

5 Conclusion

In this work, we explore the LLMs for solving the Multimodal Emotion-Cause Pair Extraction with Emotion Category (MECPE-Cat) task. Through a pilot study, we first select an LLM, ChatGLM, that assists in achieving optimal task performance. The backbone ChatGLM receives textual dialogue, and also perceives the multimodal information via the ImageBind vision encoder. Lastly, we devise an emotion-cause-aware instruction-tuning mechanism for updating LLMs, which enhances the perception of the emotions with their corresponding causal rationales. Our system achieves a weighted average F1 score of 34.71%, securing second place on the MECPE-Cat leaderboard.

Acknowledgements

This work is sponsored by CCF-Baidu Open Fund.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yuyang Chai, Chong Teng, Hao Fei, Shengqiong Wu, Jingye Li, Ming Cheng, Donghong Ji, and Fei Li. 2022. Prompt-based generative multi-label emotion prediction with label contrastive learning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 551–563. Springer.

Shunjie Chen, Xiaochuan Shi, Jingye Li, Shengqiong Wu, Hao Fei, Fei Li, and Donghong Ji. 2022. Joint alignment of multi-task feature and label spaces for emotion cause pair extraction. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 6955–6965.

Zifeng Cheng, Zhiwei Jiang, Yafeng Yin, Cong Wang, Shiping Ge, and Qing Gu. 2023. A consistent dual-mrc framework for emotion-cause pair extraction. *ACM Transactions on Information Systems*, 41(4):1–27.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023a. Reasoning implicit sentiment with chain-of-thought prompting. *arXiv preprint arXiv:2305.11255*.

Hao Fei, Qian Liu, Meishan Zhang, Min Zhang, and Tat-Seng Chua. 2023b. Scene graph as pivoting: Inference-time image-free unsupervised multimodal machine translation with visual scene hallucination. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5980–5994.

Hao Fei, Yafeng Ren, Yue Zhang, and Donghong Ji. 2023c. Nonautoregressive encoder-decoder neural framework for end-to-end aspect-based sentiment

triplet extraction. *IEEE Trans. Neural Networks Learn. Syst.*, 34(9):5544–5556.

Hao Fei, Yue Zhang, Yafeng Ren, and Donghong Ji. 2020. Latent emotion memory for multi-label emotion classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7692–7699.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018. Icon: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2594–2604.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Bobo Li, Hao Fei, Fei Li, Yuhan Wu, Jinsong Zhang, Shengqiong Wu, Jingye Li, Yijiang Liu, Lizi Liao, Tat-Seng Chua, et al. 2022. Diaasq: A benchmark of conversational aspect-based sentiment quadruple analysis. *arXiv preprint arXiv:2211.05705*.

Bobo Li, Hao Fei, Lizi Liao, Yu Zhao, Chong Teng, Tat-Seng Chua, Donghong Ji, and Fei Li. 2023. Revisiting disentanglement and fusion on modality and context in conversational multimodal emotion recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5923–5934.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, et al. 2021. Recognizing emotion cause in conversations. *Cognitive Computation*, 13:1317–1332.

Leigang Qu, Shengqiong Wu, Hao Fei, Liqiang Nie, and Tat-Seng Chua. 2023. Layoutllm-t2i: Eliciting layout guidance from llm for text-to-image generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 643–654.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Shengqiong Wu, Hao Fei, Wei Ji, and Tat-Seng Chua. 2023a. Cross2stra: Unpaired cross-lingual image captioning with cross-lingual cross-modal structure-pivoted alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2593–2608. Association for Computational Linguistics.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2023b. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*.

Shengqiong Wu, Hao Fei, Hanwang Zhang, and Tat-Seng Chua. 2024. Imagine that! abstract-to-intricate text-to-image synthesis with scene graph hallucination diffusion. *Advances in Neural Information Processing Systems*, 36.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Ao Zhang, Hao Fei, Yuan Yao, Wei Ji, Li Li, Zhiyuan Liu, and Tat-Seng Chua. 2024a. Vpgtrans: Transfer visual prompt generator across llms. *Advances in Neural Information Processing Systems*, 36.

Meishan Zhang, Bin Wang, Hao Fei, and Min Zhang. 2024b. In-context learning for few-shot nested named entity recognition. *arXiv preprint arXiv:2402.01182*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Yu Zhao, Hao Fei, Yixin Cao, Bobo Li, Meishan Zhang, Jianguo Wei, Min Zhang, and Tat-Seng Chua. 2023. Constructing holistic spatio-temporal scene graph for video semantic role labeling. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5281–5291.

Li Zheng, Donghong Ji, Fei Li, Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, and Chong Teng. 2023. Ecqed: Emotion-cause quadruple extraction in dialogs. *arXiv preprint arXiv:2306.03969*.

TueCICL at SemEval-2024 Task 8: Resource-efficient approaches for machine-generated text detection

Daniel Stuhlinger
University of Tübingen
daniel.stuhlinger@student.uni-tuebingen.de

Aron Winkler
University of Tübingen
aron.winkler@student.uni-tuebingen.de

Abstract

Recent developments in the field of NLP have brought large language models (LLMs) to the forefront of both public and research attention. As the use of language generation technologies becomes more widespread, the problem arises of determining whether a given text is machine generated or not. Task 8 at SemEval 2024 consists of a shared task with this exact objective. Our approach aims at developing models and strategies that strike a good balance between performance and model size. We show that it is possible to compete with large transformer-based solutions with smaller systems. Our code can be found on GitHub. [1]

1 Introduction

Recent developments in the field of NLP have brought large language models (LLMs) to the forefront of both public and research attention. With the introduction of GPT-3 (Brown et al., 2020), made accessible to the public through ChatGPT, the gates were open to the generation of high-quality text through AI. This leads to a sort of arms race both in integrating AI into customer-facing products, as well as in developing the models themselves, leading, among others, to Facebook's Llama (Touvron et al., 2023) and the open source model Mistral (Jiang et al., 2023).

Many fields have seen a dramatic increase in the use of LLMs, including arts, education, software development, and many more. Initial public reaction to the popularisation of LLM-powered chat interfaces already highlighted its potential problems for plagiarism detection in scientific, educational and other contexts (Dehouche, 2021). In a landmark development, the 5-month-long strike of the Writer's Guild of America[2] resulted in an agreement which included safeguards for writers against the use of artificial intelligence.[3]

The widespread use of language generation technologies is only expected to grow. However, this development is accompanied by a surging need to automate the process of flagging machine-generated text.

Crothers et al., 2023 offers a comprehensive survey of machine-generated text detection strategies. Statistical methods, with global text vector representations, were among the early strategies adopted to tackle the issue. These feature vectors include, for example, TF-IDF, frequency features investigating word or n-gram distributions, readability-related features such as the Gunning-Fog index, or linguistic features such as POS-tag distributions and coreference resolution relationships within a text. The subsequently introduced neural approaches show better performance, even when paired with the aforementioned text representation strategies. Most prominently, transformer fine-tuning has established itself as more or less of a standard, with RoBERTa (Liu et al., 2019) in particular being the most strongly represented model. Zero-shot approaches, optionally coupled with fine-tuning, have also seen experimentation, but have been observed to generalise poorly across domains.

Task 8 at SemEval-2024 (Wang et al., 2024) is a shared task built around the idea of detecting machine-generated texts across a variety of domains and setups. It is structured across three subtasks (subtask A, subtask B, subtask C), our team decided to only tackle subtasks A and C. Subtask A is a binary classification task, with the objective of determining whether a text is human- or machine-generated. The subtask has a monolingual (English) and a multilingual track - our team only

[1] https://github.com/cicl-iscl/TueCICL_SemEval2024

[2] https://www.vox.com/culture/2023/9/24/23888673/wga-strike-end-sag-aftra-contract

[3] https://www.nbclosangeles.com/news/local/hollywood-writers-safeguards-against-ai-wga-agreement/3233064/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1597–1601

submitted for the monolingual track. Subtask C is a boundary detection task: here, texts have a human segment followed by a machine-generated segment. The objective of the subtask is to correctly predict the boundary index.

Our approach for both tasks was to try to obtain competitive solutions with low resource cost. For this reason we deployed two different model classes: LSTM-based models (Hochreiter and Schmidhuber, 1997) and Multilayer Perceptrons (MLP) (Popescu et al., 2009) and relied on various strategies of representing the input texts, either at the token or at the text level. We experimented with character-level approaches, systems relying on pretrained Word2vec (Mikolov et al., 2013) embeddings, and linguistically motivated features at the text level through spaCy (Honnibal et al., 2020) and the TextDescriptives (Hansen et al., 2023) package.

2 Methods and experimental setup

2.1 Subtask A

For subtask A, our intuition was that surface-level and stylistic features would be more effective than semantics in discriminating between human and machine-generated text. To build on this idea, we developed three approaches.

The first approach involved training a character-level LSTM. We expect the stylistic features of the texts to be good indicators of the generator, and working at the character level is known to capture this information well. For example, character n-gram models have been used successfully in the field of authorship attribution, which relies heavily on style (Stamatatos et al., 2013).

First, input texts are tokenized at the character-level, all tokens are mapped to their lowercase variants, and lastly numerals and punctuation are mapped to a <NUM> and a <PUNCT> special token respectively. White-space elements (space, tab, newline) are also mapped to a special token <WS>. At this point, the tokenized and transformed inputs are fed through an LSTM, and the representation of the last token is used for prediction.

The second approach is constructed along the lines of the first in terms of technical setup, but deals with words rather than characters. Large transformer-based solutions benefit from vast amounts of pre-training, but at a heavy computational cost – using pretrained embeddings as model inputs appeared to be a good compromise between heavy models and training from scratch, as was the case with the character-level LSTM. We used the pretrained Word2vec embeddings from the Wikipedia2Vec (Yamada et al., 2020) project to map texts to vectors, but maintained the other steps, such as mapping numerals, punctuation and white-space to special tokens.

The third approach is not recurrent in its nature – instead, we used the TextDescriptives pipeline (Hansen et al., 2023) through Spacy (Honnibal et al., 2020) to obtain 66 lingustically motivated features to globally represent the text. Such linguistic features have a long tradition in NLP, for example in the field of readability analysis (for example Vajjala and Meurers, 2012), and have been observed to be valid and cheap-to-compute representations in a variety of settings. Since they are most well known for capturing the style of a text, rather than semantics, they appear to be very well suited for the present task. Additionally, we computed the mean perplexity of the document using GPT-2 (Radford et al., 2019) and added it to the feature vectors. This follows the idea that the perplexity of a document assigned by a LLM should be higher for human written texts than for machine generated texts (Chaka, 2023). Our third approach computes this global feature vector for the input text, then generates a prediction through a simple MLP. The model consists of 3 linear layers with Tanh activation functions in between and was trained for 2000 epochs with a learning rate of 0.0003.

Lastly, we formulated a joint model which takes as input the final representations (the last hidden states) of each of the three previous approaches, thus generating a single prediction.

Model	Type	L*	H**
Character-level	LSTM	2	512
Word2vec	LSTM	2	512
Language features	MLP	3	256
Joint model	FFN	-	512

Table 1: Summary of models for subtask A. * Number of layers. ** Hidden size.

2.2 Subtask C

For subtask C, which required predicting the boundary position in texts between a human and a machine-generated segment, we adopted the same guiding principles in developing our solutions as

we had done in subtask A. The overarching objective was to arrive at competitive models while remaining within a certain size constraint.

An additional challenge for this subtask, aside from an increase in difficulty in the objective itself, was the relative scarcity of training data. While in subtask A the training set had over one hundred thousand records, the training material in subtask C consisted of just below four thousand texts. As such, any strategy that did not involve pretraining would be severely disadvantaged in this setting.

Our first approach for this subtask was also a character-level solution, with the same general setup as in subtask A. For generating a prediction, however, the representations at every token are evaluated, and a classification is performed. In this sense, the model is predicting whether any given character is in the human or the machine-generated segment of the text. The first token predicted to be machine-generated is taken to be the boundary position. Importantly, the LSTM we implemented for this subtask is bidirectional, meaning at every token the model has awareness of both the left and right contexts.

To offset the relative lack of training items, together with the absence of any model pre-training in this case, we trained this model on both subtask A and subtask C data for 5 epochs, then trained further on only task C data for 3 further epochs. This improved performance significantly on the development set.

We also implemented a Word2vec solution along the lines of what was described for subtask A. We opted for a bidirectional LSTM, and enhanced the training data as described earlier, though the effects of this were less prominent owing to the use of pretrained embeddings.

We also built a joint model over the aforementioned character- and Word2vec models. This consisted in a FFN whose inputs were the concatenated representations at the word level. For the character-level model, this meant averaging the representation at every character for any given word.

Model	Type	L*	H**
Character-level	LSTM	2	512
Word2vec	LSTM	2	512
Joint model	FFN	-	256

Table 2: Summary of models for subtask C. * Number of layers. ** Hidden size.

3 Results

3.1 Subtask A

Model	Dev	Test	Ranking
Baseline	0.72	0.88	20
Character-level	0.85	0.55	127
Word2vec*	0.82	0.72	85
Language features*	0.63	0.88	21
Joint model*	0.83	0.69	96

Table 3: Results for SemEval-2024 Task 8, subtask A. Dev and Test columns report the accuracy on the respective data partitions. The ranking column refers to the model ranking in the shared task competition. The scores and ranking of the unofficial submissions were not provided by the organisers and computed by us. There was a total of 137 submissions.
* unofficial submissions

Table 3 shows the results for each model on subtask A. On the development set, almost all models outperform the transformer baseline provided by the organisers. The best performing model was the character-level model, with an accuracy of 0.85 – this was our final submission for the shared task.

While the two recurrent models and the joint model do not differ very much from one another, the FFN built on linguistically motivated global feature vectors sets itself apart in that it is the worst performing model on the development set.

The character-level model only achieves an accuracy of 0.55 on the test set, while the Word2vec and joint models achieve 0.72 and 0.69 respectively – all falling short of the baseline. Surprisingly, the language feature model is head and shoulders above the rest when it comes to the test set, with an accuracy of 0.88 that matches the baseline. Our assumption is that this is due to a conceptual difference between development and test set. A possible reason could be that the domain for human-written texts in the test set were student essays only. The development set on the other hand consisted of human-written texts from multiple domains. The linguistic features prevalent in the student essays seem to be more distinctive for classifying the documents compared to the texts from multiple domains.

3.2 Subtask C

Table 4 outlines our results for subtask C. In this subtask, we were unable to match the transformer baseline.

Model	Dev	Test	Ranking
Baseline	3.53	21.54	14
Character-level*	8.35	45.83	28
Word2vec*	7.02	38.35	27
Joint model	6.36	34.88	25

Table 4: Results for SemEval-2024 Task 8, subtask C. Dev and Test columns report mean absolute error (MAE) on the respective data partitions. The ranking column refers to the model ranking in the shared task competition. The scores and ranking of the unofficial submissions were not provided by the organisers and computed by us. There was a total of 33 submissions.
* unofficial submissions

Our official submission, the joint model, achieved a mean standard error of 6.36 on the development set, falling short of around 3 points from the baseline provided by the organisers. The difference is even more dramatic when it comes to the test set, where the gap widens to around 13 points. This is also far off from the best performing solutions in the shared task, which achieved a MAE of 15.7.

The character and Word2vec models failed to outperformed both the baseline and the joint model for the development set, and this remains the case in the test set. This reinforces the idea that extracting as much information as possible from the texts is key to performance in this subtask.

Overall, the models developed for subtask C have proven to be somewhat unrefined. The test set seems to be particularly punishing towards solutions that do not generalize well, but the results, while highlighting the shortcomings of our models, also point toward the potential that these approaches can have, with more attention dedicated to them.

4 Conclusion and Discussion

Our objective for Task 8 at Semeval-2024 was to compete with large-scale solutions with models that could run on commonplace systems like mid-range laptops. For this purpose, we discarded LLM-based solutions that are prevalent in the related work in the field, opting instead for LSTMs and MLPs whose size can more easily be controlled.

While our team did manage to keep model size under control (none of the proposed solutions require more than 1 GB of memory), the systems we proposed performed less than ideally in the task itself. On both subtasks we participated in, our best models failed to match the transformer baseline in the test set, despite positive results in the development set.

Despite the final results, we believe our approach to be valid. Our development processes likely ended up producing models that were overly tuned to the development set. With more time, it would be possible to produce more refined and generalizable solutions. Trying different training strategies, like contrastive learning, or different architectures, such as mixture-of-experts systems, might be a good direction to follow in future work.

To most problems that arise in the field of NLP, researchers and companies increasingly respond with huge models that require dedicated servers to run. But for many users, keeping their data safely on their own machines is a priority, thus discarding many of the contemporary LLM-based services. System designers should aim to strike a compromise between size and performance, and prioritise users being able to own their workflows when possible. Like our attempt did for this shared task, we believe researches should consider these objectives when proposing new solutions.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Chaka Chaka. 2023. Detecting AI content in responses generated by ChatGPT, YouChat, and Chatsonic: The case of five AI content detection tools. *Journal of Applied Learning and Teaching*, 6(2).

Evan N. Crothers, Nathalie Japkowicz, and Herna L. Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002.

Nassim Dehouche. 2021. Plagiarism in the age of massive generative pre-trained transformers (GPT-3). *Ethics in Science and Environmental Politics*, pages 17–23.

Lasse Hansen, Ludvig Renbo Olsen, and Kenneth Enevoldsen. 2023. TextDescriptives: A Python package for calculating a large variety of metrics from text. *Journal of Open Source Software*, 8(84):5153.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. 2009. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 7(1):579–588.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ph D Stamatatos et al. 2013. On the robustness of authorship attribution based on character n-gram features. *Journal of Law and Policy*, 21(2):7.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 163–173, Montréal, Canada. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. SemEval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30. Association for Computational Linguistics.

GeminiPro at SemEval-2024 Task 9: BrainTeaser on Gemini

Kyu-Hyun Choi and **eung-Hoon Na**
Division of Computer Science and Engineering, Jeonbuk National University, South Korea
chlrbgus321@naver.com and nash@jbnu.ac.kr

Abstract

It is known that human thought can be distinguished into lateral and vertical thinking. The development of language models has thus far been focused on evaluating and advancing vertical thinking, while lateral thinking has been somewhat neglected. To foster progress in this area, SemEval has created and distributed a brainteaser dataset based on lateral thinking consist of sentence puzzles and word puzzle QA. In this paper, we test and discuss the performance of the currently known best model, Gemini, on this dataset.

1 Introduction

Human thought is known to be distinguished into lateral and vertical thinking. (Jiang et al., 2023) cites (Waks, 1997) in mentioning, based on modern neuroscience, that vertical thinking is associated with the left hemisphere of the brain, while lateral thinking is associated with the right hemisphere. Moreover, this paper notes that during the development of language models, there has been a focus on problem-solving abilities in vertical thinking, neglecting the capabilities based on lateral thinking. This paper anticipates that lateral thinking puzzles may not be easily solved with just additional adaptations and extensions of the LLM (Large Language Model) approach, yet this paper is prepared to counter that expectation. It evaluates the performance of Google's ambitious model, Gemini(Team et al., 2023), which is said to surpass GPT-4, by measuring performance solely through changes in demonstration, as it was not possible to fine-tune Gemini.

Gemini is anticipated to show increased performance due to the scaling law mentioned in (Kaplan et al., 2020), as it utilizes significantly more parameters than GPT-4. Being a multimodal model trained with additional learning resources such as visual and auditory inputs, it is speculated that these characteristics might give rise to unique emergent abilities. (Wei et al., 2022) These two aspects are expected to contribute to performance improvements in lateral thinking.

Our approach is straightforward. First, we formalize Gemini's responses by adding demonstrations, following the same few-shot provision method as used by SemEval, and second, we provide only the relevant task few-shot examples for the two brainteaser tasks: sentence puzzles and word puzzles. Through the second method, we investigate whether providing clear few-shot examples for tasks alone can aid in performance improvement.

2 Background

2.1 Vertical thinking

As illustrated in Figure 1 of the (Jiang et al., 2023), vertical thinking is generally considered a logical form of thought. The first example of vertical thinking in Figure 1 of the paper, regarding the question "How do you flood a room?", involved associating the meaning of the word "flood" with the span "Cover with water". This association led to the selection of a similar meaning, "Fill it with water", as the answer. The second example of vertical thinking was in response to the question "I have five fingers, but I am not alive. What am I?". Here, the span "five fingers" led to the association of a similar span "Five separate parts", and "Not alive" led to the association of "item like a hand", which, despite "not alive" having a broader meaning, was contextually restricted by the span "five fingers". The only option that simultaneously had the properties of "Five separate parts" and "item like a hand" was "Glove". This problem, even though it is a riddle as mentioned in brainteasers, could be solved through vertical thinking. Such an ability to associate a specific word span with another span of similar meaning could be implemented in transformer

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1602–1606

models, as mentioned in (Dai et al., 2021), where the feed-forward network contains knowledge, and the context patterns created in the attention layer act as a key, enabling the association of a particular part of the input with another similar span.

2.2 Lateral thinking

Let's look at the first example of lateral thinking from Figure 1 of the overview paper. It is common sense to associate "Man shaves everyday" with "His beard gets clean everyday". However, the condition "yet keeps his beard long" blocks this inference path. Therefore, the model must use a different reasoning path, and to solve the problem, it must break away from the common sense that the man shaves himself and instead think of the possibility that he shaves someone else. This example forces the most commonsensical reasoning path to be blocked and requires navigating an alternative reasoning path.

The second example asks, "What type of cheese is made backwards?" This question is not commonsensical in itself. However, if "made" is not considered as a verb but as a sequence of letters, the problem is solved. Reversing "made" spells "edam," which is a type of cheese.

2.3 Brain-Teaser Benchmark

Brain teaser tasks (Jiang et al., 2024) are designed to explore whether language models are capable of lateral thinking, diverging from traditional methods. These tasks involve reading a question and providing an answer in a QA format, structured as a multiple-choice question with options (A), (B), (C), (D) to ensure clear output.

Sentence puzzles involve semantic exercises that break conventional thinking, while word puzzles use arrangements of alphabets in words to provide answers that play on words, challenging common sense.

There are two variations of both sentence and word puzzles. One is semantic reconstruction, where the question is paraphrased to measure if the problem can still be solved effectively while the answer and options remain unchanged. The other is context reconstruction, where the thought process to solve the problem remains the same, but the question and options are changed.

Two methods are used to measure performance: instance-based accuracy, which measures the accuracy of original, semantic, and context reconstructions separately, and group-based accuracy, which increases accuracy if the original and semantic reconstructions are answered correctly together or if correct answers are provided for original, semantic, and context reconstructions all at once.

Approximately 1,000 training examples were provided by Semeval, but this study measures the intrinsic ability of Gemini without using the training dataset.

3 System overview

In this study, we used Gemini, an ambitious model released by Google, known to surpass ChatGPT. We utilized the Gemini-Pro API and followed the ChatGPT evaluation method provided by SemEval.

3.1 Add Demonstration

Gemini tends to include explanations in its responses, resulting in varying output styles for each question. For example, it can be feel like this:

The answer is (A), because [explanation.....]

[explanation.....] so, the answer is (A)

(B) is [explanation.....]
(C) is [explanation.....]
(D) is [explanation.....]
so, the answer is (A)

To use the brain teaser score calculator, the answers must be clear in the form of (A), (B), (C), or (D). The output style described above is not suitable for input into the answer calculator, especially in the last example where all options (A), (B), (C), and (D) are included in the output. Implementing an algorithm to post-process this and select a clear single answer, like (A), from such outputs is complex. Therefore, to avoid these difficulties, we structured the demonstration to include the following feel.

[demonstration...]
question
option (A)
option (B)
option (C)
option (D)

3.2 Use only relevant few-shot examples

To determine if providing only sentence puzzle examples for sentence puzzles or only word puzzle

examples for word puzzles helps resolve confusion between examples and aids in problem-solving, we conducted 1-shot, 2-shot, and 4-shot evaluations using the same set of examples.

In this case, we did not add a demonstration because the few-shot examples clearly provide the style of output. When using only relevant few-shot examples, we follow this format:

For sentence puzzles:

N examples
[sentence puzzle question
option (A)
option (B)
option (C)
option (D)
Answer: (A) or (B) or (C) or (D)]

problem
[sentence puzzle question
option (A)
option (B)
option (C)
option (D)
Answer:]

For word puzzles:

N examples
[word puzzle question
option (A)
option (B)
option (C)
option (D)
Answer: (A) or (B) or (C) or (D)]

problem
[word puzzle question
option (A)
option (B)
option (C)
option (D)
Answer:]

4 Experimental setup

Although SemEval provided aproxymately 1,000 training examples, this study did not use the training data as it did not involve fine-tuning. Instead, we directly used the brain teaser test data to measure the intrinsic capabilities of Gemini-Pro. Gemini-Pro occasionally does not output an answer. In such cases, we considered (D) as the answer. If it does not output a response in the structured form of (A), (B), (C), (D), we also treated it as (D). For all other cases, we followed the ChatGPT methodology as outlined by SemEval.

5 Results

5.1 With Demonstration

As shown in table 1, for zero-shot, sentence puzzle performance was generally superior to chatGPT, except it showed exceptionally lower performance in context reconstruction. In few-shot, when two examples were provided, it only showed superiority in original, and tied with four-shot in ori&sem&con, while four-shot generally showed superior performance elsewhere, and performance actually decreased in eight-shot.

For word puzzles, zero-shot performance was superior to chatGPT in original, semantic, and context, but uniquely showed lower performance in Ori&sem and ori&sem&con. In few-shot, original showed overwhelming performance in two-shot, semantic was superior in eight-shot, and context had the best performance in four-shot. Overall, the best performance was seen in eight-shot.

5.2 Without Demonstration and Use only relevant few-shot examples

When only sentence puzzle examples were provided for sentence puzzles, the performance in two-shot and four-shot was comparable to the original method. In two-shot, original performance dropped by 10 points, semantic increased by 8 points, context increased by 3 points, ori&sem dropped by 3 points, and ori&sem&con dropped by 5 points, with overall scores remaining the same. In four-shot, original remained unchanged, semantic dropped by 3 points, context dropped by 10 points, Ori&sem increased by 8 points, and ori&sem&con remained the same, with overall dropping by 5 points.

For word puzzles in two-shot, original dropped by 10 points, semantic increased by 6 points, context increased by 4 points, but uniquely, performance remained the same in ori&sem and ori&sem&con, with overall performance unchanged. In four-shot, original performance increased by 10 points, semantic by 19 points, context dropped by 3 points, ori&sem increased by 16 points, and ori&sem&con by 16 points, with an

	Instance-based			Group-based		Overall
	Original	**Semantic**	**Context**	**Ori & Sem**	**Ori & Sem & Con**	
With Demonstration						
Sentence Puzzle						
Zero-shot	0.67	0.62	0.62	0.52	0.4	0.64
Two-shot	0.75	0.67	0.72	0.6	0.57	0.71
Four-shot	0.72	0.7	0.75	0.67	0.57	0.72
Eight-shot	0.7	0.67	0.67	0.57	0.45	0.68
Word Puzzle						
Zero-shot	0.65	0.53	0.53	0.43	0.25	0.57
Two-shot	0.78	0.78	0.71	0.5	0.40	0.69
Four-shot	0.69	0.56	0.87	0.43	0.40	0.69
Eight-shot	0.71	0.71	0.84	0.59	0.53	0.76
Without Demonstration and Use only relevant few-shot examples						
Sentence Puzzle						
One-shot	0.72	0.72	0.62	0.65	0.52	0.69
Two-shot	0.65	0.75	0.75	0.57	0.57	0.71
Four-shot	0.72	0.67	0.65	0.65	0.57	0.67
Word Puzzle						
One-shot	0.75	0.59	0.78	0.78	0.56	0.70
Two-shot	0.68	0.65	0.75	0.5	0.40	0.69
Four-shot	0.75	0.75	0.84	0.59	0.56	0.76

Table 1: Result of evaluation on Gemini-Pro

overall increase of 7 points.

Sentence puzzles showed a tendency for scores to drop, regardless of how the examples were organized, making it unclear whether the scores dropped randomly. Word puzzles showed a tendency for significant performance increases, but with only 96 test examples for word puzzles and no clear direction in the fluctuations of scores, it is uncertain whether the performance increase was due to providing only word puzzle examples or if the performance randomly improved.

6 Conclusion

As observed in Figure 2 of (Jiang et al., 2023), increasing the number of examples in sentence puzzles did not consistently improve performance, and while an overall upward trend in performance for word puzzles was noted, it did not improve regularly. Similarly, in the experiments of this paper, performance fluctuations with the number of examples were erratic, but it is clear that performance is generally higher compared to chatGPT. The leaderboard for brainteasers often shows many cases scoring over 90, which is likely due to the use of fine-tuning methods on the brainteaser training set. Without training specialized for brainteasers, the effect of using the method of demonstration appears to be minimal or almost nonexistent in a pure model state, and it has been found that larger models exhibit more pronounced performance improvements. Particularly, Gemini, despite being a multimodal model trained with both visual and auditory inputs, significantly underperforms compared to human capabilities. Contrary to the original paper's expectation, it was observed that merely increasing the model size could spontaneously develop problem-solving abilities for lateral thinking tasks, suggesting that even the capability for lateral thinking falls within the range of emergent abilities. According to (Jawahar et al., 2019), as the training of transformer models progresses, layers specialized for tasks are formed, with it being speculated that lateral associations are made in highly differentiated semantic layers in layers closer to the end. Perhaps the improvement in performance in LLMs, as mentioned in the context of brainteasers, might simply be due to memorizing content from the corpus.(Carlini et al., 2022) It remains to be seen whether probing layers specialized for semantic tasks in the future could unveil the mechanism behind lateral thinking.

7 Acknowledgement

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub)

References

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *CoRR*, abs/2104.08696.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6:245–255.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Archimedes-AUEB at SemEval-2024 Task 5: LLM explains Civil Procedure

Odysseas S. Chlapanis[1,3], **Ion Androutsopoulos**[1,3] and **Dimitrios Galanis**[2,3]
[1]Department of Informatics, Athens University of Economics and Business, Greece
[2]Institute for Language and Speech Processing, Athena Research Center, Greece
[3]Archimedes Unit, Athena Research Center, Greece
odyhlapanis@aueb.gr, ion@aueb.gr, galanisd@athenarc.gr

Abstract

The SemEval task on Argument Reasoning in Civil Procedure is challenging in that it requires understanding legal concepts and inferring complex arguments. Currently, most Large Language Models (LLM) excelling in the legal realm are principally purposed for classification tasks, hence their reasoning rationale is subject to contention. The approach we advocate involves using a powerful teacher-LLM (ChatGPT) to extend the training dataset with explanations and generate synthetic data. The resulting data are then leveraged to fine-tune a small student-LLM. Contrary to previous work, our explanations are not directly derived from the teacher's internal knowledge. Instead they are grounded in authentic human analyses, therefore delivering a superior reasoning signal. Additionally, a new 'mutation' method generates artificial data instances inspired from existing ones. We are publicly releasing the explanations as an extension to the original dataset, along with the synthetic dataset and the prompts that were used to generate both. Our system ranked 15th in the SemEval competition. It outperforms its own teacher and can produce explanations aligned with the original human analyses, as verified by legal experts.

1 Introduction

Semeval-2024 Task 5 (Held and Habernal, 2024) concerns Legal Reasoning within the realm of US Civil Procedure, based on the *Legal Argument Reasoning Task in Civil Procedure* dataset (Bongard et al., 2022). It requires understanding legal concepts and advanced reasoning capabilities, such as the ability to grasp analogy-based arguments and identify contradictions. It is cast as a binary classification problem where, given a question and a candidate answer, the system has to respond whether the candidate answer is correct or not. The questions and answers originate from a textbook widely used by US law schools.

Introduction
... Klaxon: a federal diversity court should use the choice-of-law rules of the state in which it sits. ...
Question
A Rhode Island citizen goes skiing in Vermont. He falls and gets injured. He brings a diversity action against the operator in federal court in Rhode Island. ... The judge would probably apply
Answer (label: *correct*)
Vermont law, because the accident happened there.
Expert Analysis
Rhode Island's choice-of-law rule calls for application of the law of the place of injury. ...
Model Explanation
The federal district judge would probably apply Vermont law, as per Klaxon holding that the judge should use the local state's choice-of-law rules.

Table 1: Example of a training instance and a system-generated explanation aligned with the expert's analysis.

Our system leverages two data augmentation strategies to enrich the provided training set. The first strategy involves extending each data point with Chain-of-Thought (Wei et al., 2023) style explanations originating from the author's original analysis and refined by GPT-3.5. The second strategy generates additional synthetic examples inspired by the original training examples. The synthetic examples are also accompanied by explanations. We call this method *data mutation*. The data generated from GPT-3.5 for both methods and the prompts used are publicly available.[1]

We fine-tune a Llama-2 (Touvron et al., 2023b) model using both synthetic and original data, incor-

[1]github.com/nlpaueb/multiple-choice-mutation

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1607–1622

porating explanations, to develop a model capable of generating responses and explaining the reasoning supporting them. Our findings indicate that both data augmentation methods significantly improve the model's performance. We conducted ablation studies to analyze the effects of each method. In addition, legal experts evaluate our model's explanations manually to offer us valuable insights about the quality of the produced explanations and the root causes of its errors.

2 Background

2.1 Task Setup

The dataset used in SemEval-2024 Task 5 is derived from a textbook on US civil procedure (Glannon, 2018). Each data instance consists of three parts: introduction, question, and answer (with the label of the answer), as shown in Table 1. The training and development instances also have an additional section, containing an analysis of the textbook's author (see 'expert analysis' in Table 1). The goal is to predict the label (correct or incorrect) of the given answer. The latter has the form of a completion of an incomplete text (the 'question' part of the instance). Each chapter of the book addresses a specific topic, which is introduced in the 'introduction' of each instance.

2.2 Related Work

In the last few years, several models were proposed for legal tasks. LegalBERT (Chalkidis et al., 2020), a BERT-based model (Devlin et al., 2019) that has been further pre-trained on legal data, achieved state-of-the-art results in three downstream tasks. However, LegalBERT is unable to generate explanations for its decisions. In subsequent work to address this limitation, a new hierarchical extension of LegalBERT was proposed (Chalkidis et al., 2021) that can select paragraphs of the input texts that justify its decisions, but it has to be trained with annotated data. Lawyer LLaMA (Huang et al., 2023) enhanced the Chinese LLaMA (Cui et al., 2023), a descendant of LLaMA-1 (Touvron et al., 2023a), with legal knowledge. This was achieved by supervised fine-tuning on synthetic or manually created legal Chinese datasets.

Chain-of-Thought (CoT) (Wei et al., 2023) was introduced as "a series of intermediate natural language reasoning steps that lead to the final output". Few-shot CoT prompting (Wei et al., 2023) is a method that enhances the reasoning skills of LLMs by providing a few CoT demonstrations as exemplars in the prompt before asking the LLM for the final answer. Furthermore, it has been shown that CoT prompting can be improved by using a technique called self-consistency decoding (Wang et al., 2023), where sampling is used during decoding to obtain multiple answers, and then the most consistent one is chosen (e.g., via majority voting).

Instruction tuning (Ouyang et al., 2022) fine-tunes an LLM on a wide range of input requests and the desirable responses, in order to train the LLM to follow instructions. WizardLM (Xu et al., 2023) implements instruction tuning on synthetic instruction data of progressively increasing complexity. The synthetic instruction data are commonly generated by a powerful teacher-LLM. Reinforced Self-Training (ReST) (Gulcehre et al., 2023) iteratively enhances synthetic data by automatically filtering out lower-quality samples and then utilizing the remaining synthetic data to improve the model that produces the synthetic data. Orca (Mukherjee et al., 2023) introduces explanation tuning, which incorporates fine-tuning on CoT data generated by GPT-4 to teach advanced reasoning skills to a smaller LLM.

3 System overview

3.1 Method

Our approach is based on a small open-source LLM (LLama-2-7B), which is fine-tuned to generate CoT-like explanations supporting its predictions. A much larger teacher LLM (GPT-3.5) is utilized to acquire appropriate reasoning data (CoT explanations) with two data augmentation techniques. The first technique modifies the experts' analyses (Section 3.3) to turn them into CoT explanations appropriate for fine-tuning. The second technique 'mutates' an original training instance to generate a synthetic one that incorporates alternative fictitious elements, but requires the application of similar legal reasoning to arrive at the correct answer (Section 3.4). Experiments show that both techniques boost the performance of the baseline model, which is the same LLM fine-tuned without explanations.

3.2 Prompt structure

To create appropriate reasoning data, GPT-3.5 has to perform challenging tasks and the key to achieve this is handcrafting clear and concise prompts. We follow prompt-engineering best practices (Bsharat et al., 2024) to create prompts for both data aug-

mentation techniques. Each prompt has three parts: system instructions, an example of a query and an appropriate response, and a new query that GPT-3.5 has to respond to as in the example. The system instructions are designed to a) delineate the model's role, typically attributed as a legal expert in US Civil Procedure, b) outline the task to be executed, and c) specify the desired output format. Complete examples are provided in Appendix A.1.

3.3 Human-guided explanations (HGE)

CoT explanations that will be leveraged for fine-tuning must adhere to a uniform and formal style. Furthermore, we desire our model's explanations to be succinct so that reasoning fallacies can be easily detected by legal experts. Although the expert analysis (Table 1) provided by the dataset is similar to a CoT explanation, it is tailored for students and therefore more detailed and informal. We refine the expert's analysis with the assistance of GPT-3.5 (Table 2 shows the prompt used) to align it with our desired properties.

HGE
Input: Training instance **Prompt:** Explain why the answer is correct/incorrect according to the analysis. **ChatGPT's response:** Explanation: ...

Table 2: The Human-Guided Explanations (HGE) prompt asks the LLM to provide a CoT explanation that aligns with the analysis of the legal expert and follows a specific format.

3.4 Multiple choice mutation (MCM)

Our goal with this technique is to generate artificial data that demand similar reasoning skills and legal knowledge to the original ones. You can see an example in Table 3 for the training instance in Table 1. We use GPT-3.5 as the teacher-LLM. We implement this in two different prompting stages.

In the first prompting stage (see Table 4), we provide an instance from the training data (introduction, question and answer) and ask GPT-3.5 to generate a multiple choice question with four options. Initially, the responses were often not satisfactory (you can find more details in Appendix A.2). To improve them we introduced two prompt engineering artifacts, which we call 'concept' and 'question background', that clarify the type of questions we want to generate. These artifacts are demonstrated in the example provided with the prompt. The 'concept' aims in identifying the legal outcome of the case and the 'question background' aims in generating a different fictional scenario that is not related to the original one. The final prompt asks for a multiple choice question, with a different 'background' than the original question, that illustrates the same 'concept'. We then concatenate

MCM example
Concept: Choice of Law in Civil Procedure **Question Background:** Alex, a Florida resident, buys a rare art piece from Carter, a California resident. The contract stipulates that any disputes arising from the agreement will be resolved in Arizona. A disagreement arises over the authenticity of the artwork. Alex sues Carter in federal court in Florida. ... **Multiple Choice Question:** Which state's contract law would most likely be applied in Alex's case against Carter? **Options**: A) California (label: *incorrect*) Explanation: Carter's residence is not a determining factor as the contract specifies the choice of law. B) Arizona (label: *correct*) Explanation: The contract explicitly states that any disputes will be resolved in Arizona. C) Florida (label: *incorrect*) Explanation: ... D) New York (label: *incorrect*) Explanation: ...

Table 3: Example of the MCM algorithm applied on the training instance shown in Table 1. Each option becomes a distinct mutated instance.

MCM Stage A
Input: Training instance **Prompt:** Generate a multiple choice question that illustrates the same concept with a different question background. **ChatGPT's response:** Concept Mutated Question (Question Background + Multiple Choice Question) Options: a, b, c, d Correct option: *X*

Table 4: In stage A of MCM, we generate a synthetic example question and four options, one of which is correct.

MCM Stage B
Input: Mutated question and options **Prompt:** Choose the correct option and provide an explanation for each option. **ChatGPT's response:** Correct option: *Y* Explanation for a: ... Explanation for b: ... Explanation for c: ... Explanation for d: ... **Filtering:** If X=Y, an instance for each option is created.

Table 5: In Stage B of MCM, the teacher-LLM predicts the correct option again and generates explanations for each option.

the 'question background' and the 'multiple choice question' that are both generated by the model to get the 'mutated question' that we will use as synthetic data. We discard the 'concept'. In the same prompt we also ask for the correct answer out of the candidate options.

For the second prompting stage (see Table 5) we provide as input the output of the first prompting stage without the correct answer (concept, mutated question, candidate answers). We ask again GPT-3.5 to choose the correct choice and we also ask for an explanation that justifies each choice as correct or incorrect. To avoid introducing noisy synthetic training instances, if the option chosen in the first stage is not same as the option chosen in the second stage, meaning that GPT-3.5 answered inconsistently this question, the example is discarded. We call this process the consistency filter, as it is inspired from the self-consistency approach of (Wang et al., 2023). Around 30% of the generated examples were discarded.

4 Experimental setup

4.1 Data

The initial data splits provided for the competition included a train split of 666 samples, a development (dev) split of 84 samples, and a test split of 98 samples. Out of the 84 samples of the development set, only 17 were labeled as correct. We found that these were not enough for our qualitative analysis (Section 5.6) and for this reason, we expanded the dev set by including 101 samples from the training set, resulting in 185 samples and a reduced training split of 565 samples. (We cannot conduct qualitative evaluation on the test set, as we do not possess the expert's analysis for those instances.) The F1 score is the official evaluation measure due to the dataset's imbalance, with accuracy serving as the secondary metric.

4.2 ChatGPT Prompting setup

All of our data augmentation was conducted with the model `gpt3.5-turbo-1106`. The total cost for data augmentation remained under $20. We also use GPT models with few-shot prompting as baselines. For the experiments listed in Table 6, the scores on the development set were averaged over three different random seeds for GPT-3.5 and one seed for GPT-4. In all experiments, the default OpenAI parameter values were kept.

4.3 Llama-2-7B Tuning setup

As the student model, we employed the 8-bit quantized version of the `Llama-2-7b` foundational model. We trained with QLoRA (Dettmers et al., 2023) for one epoch, with a batch size of 4, and a learning rate of 1e-4. We utilized the Hugging Face Transformers library and the Llama recipes from Facebook Research.[2] We used a single NVIDIA GPU A6000 with 48GB of GPU RAM.

5 Results

5.1 Main Results

Our best method ranked 15th among 20 competitors. However, unlike other models, it can generate concise explanations to justify its answers. Our main baseline is a Llama-2 model (Llama-2-base) fine-tuned on the (reduced) training set without data augmentation. Extending the dataset with human-guided explanations (Llama-2-HGE, Section 3.3) greatly improves performance (Table 6) while providing insightful explanations as a byproduct. Employing additional synthetic data generated by the multiple choice mutation method (Llama-2-MCM, Section 3.4) along with the human-guided explanations further enhances performance. Additionally, Llama-2-MCM outperforms GPT-3.5 prompted with CoT examples (GPT-3.5-CoT), a strong baseline that can also provide explanations. This is a notable achievement for several reasons. First of all, GPT-3.5 produced the data that were used for fine-tuning Llama-2-MCM. Furthermore, it is a much more capable few-shot reasoner than Llama-2-7B (Zheng et al., 2023) and it is a significantly larger model in terms of parameters (probably more than ten times larger, but the exact number is unknown).

5.2 BERT models

We report BERT scores (Table 6) from the work that introduced the dataset (Bongard et al., 2022). BERT-base without legal-specific pretraining achieves mediocre performance, while LegalBERT with legal pretraining is exceptional, as it would rank 8th in the competition and has similar results to a (prompted) GPT-4, a much larger (but not fine-tuned) model. However, as argued by Bongard et al. (2022), LegalBERT's responses are of limited utility, as they do not provide explanations of the reasoning behind them.

[2] github.com/facebookresearch/llama-recipes

Model	F1 (dev)	Acc (dev)	F1 (test)	Acc (test)
BERT-base (F)	-	-	56.80	**80.22**
LegalBERT (F)	-	-	65.73	76.92
GPT-3.5-base (P)	31.92	32.97	27.60	30.61
GPT-3.5-CoT (P)	55.29	59.10	43.81	45.92
GPT-4-CoT (P)	**70.43**	**75.00**	**65.88**	72.45
Llama-2-base (F)	36.13	36.24	31.25	35.33
Llama-2-HGE (F)	50.88	65.08	50.00	59.18
Llama-2-MCM (F)	55.87	66.70	51.43	61.22

Table 6: Results for BERT-based models, prompted GPT models, and Llama fine-tuned models. (F) stands for fine-tuning and (P) stands for few-shot prompting.

5.3 OpenAI GPT models

We evaluate GPT-3.5 and GPT-4, which are powerful general-purpose models, prompted with appropriate system instructions and one-shot examples. GPT-3.5-base, without CoT, acting only as a classifier model (no explanations) that predicts the correct label, performs poorly in both dev and test set (Table 6). Leveraging CoT (GPT-3.5-CoT) shows better performance, but still underperforms BERT-base and our Llama-2-MCM in the test set. GPT-4-CoT outperforms GPT-3.5-CoT (by a large margin) and has similar performance with LegalBERT (would rank 8th as well); in addition it can also produce explanations. Interestingly, the performance of GPT-3.5-CoT and GPT-4-CoT drops substantially in the test set if compared to the dev set. The decrease is larger in the case of GPT-3.5-CoT, probably because it is a less powerful model. Note that the test set is harder according to Bongard et al. (2022).

The downsides of OpenAI models are that they are costly and cannot be fine-tuned in the specific domain as easily as open-source models.

5.4 Llama models

Our main baseline, Llama-2-base, is fine-tuned on the training set without data augmentation. It produces the predicted label only, without an explanation. In a similar fashion to its GPT-counterpart (GPT-3.5-base), it performs poorly, which highlights the importance of using CoT data either in a prompting or a fine-tuning setting. Llama-2-HGE (Section 3.3), fine-tuned on the original data extended with explanations created by GPT-3.5 according to the expert's analysis, outperforms GPT-3.5-CoT on the test set. Llama-2-MCM (Section 3.4), fine-tuned on HGE data along with artificial data generated by our 'mutation' method, prevails over both Llama-HGE and GPT-3.5-CoT in every metric. It falls short of BERT-base's performance. The boost in performance that further pretraining provides to LegalBERT might suggest that such an approach (domain adaptation) would benefit our model substantially as well.

5.5 Ablation Study

In the first rows of Table 7, we report an experiment without the expert's analysis (no anls) to assess the effects of human guidance (Section 3.3). Instead, we rely on CoT explanations from GPT-3.5's inherent knowledge, following Mukherjee et al. (2023). The addition of the expert's analysis slightly improves performance, but for a more comprehensive assessment of the analysis's impact, manual evaluation of the explanations is necessary. However, we defer this aspect to future research. We also evaluate the impact of MCM's filtering process (Section 3.4). MCM's performance drops significantly (4 percentage points) without consistency filtering (no fltr, see Table 7), highlighting its value in discarding poor synthetic examples.

The plot in Fig.1 shows the effect of using more synthetic data from MCM. In the x-axis, we can see how many additional synthetic MCM instances are used for fine-tuning; the denominator is the number of synthetic data generated and the nominator (184, 344, 561, 709) are those kept after filtering. After all of the 565 original instances are utilized once, we add a second generation of synthetic data (red$^{\triangle}$ points) with a different random seed. The second generation decreases performance, which may be attributed to the addition of two synthetic instances (that describe the same legal outcome) per original instance, potentially causing overfitting.

Model	F1 (dev)	Acc (dev)
No anls	49.99	60.76
HGE	50.88	65.08
No fltr	51.14	69.41
MCM	55.87	66.70

Table 7: The effects of the expert's analysis (anls) and the consistency filter (fltr).

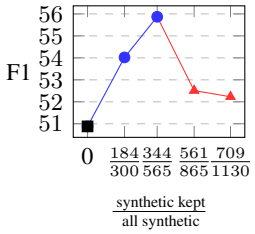


Figure 1: Llama-2-MCM scores in dev set. More mutated examples from the same GPT-3.5 query (red$^{\triangle}$) do not improve F1 score.

5.6 Qualitative Analysis

We investigate how often the model fails to generate an accurate explanation, but nevertheless succeeds to predict the correct answer-label. For this we asked two legal experts (a professor in Law and a J.D. candidate) to annotate whether the explanations of 16 correct predictions of our Llama-2-MCM model align with the expert's analysis or not (Figure 2). 8 of them were 'aligned' and the other 8 were 'not aligned'. This is an indication that performance can be deceiving, because a model might perform well without accurate reasoning. In Table 8 you can see an example that is 'not aligned' due to a deficiency in legal knowledge.

The same experts were also asked to assess 16 false predictions of Llama-2-MCM in terms of 'clarity', i.e., clear vs. unclear (see Fig.3) and comment on their observations. The objective of the second experiment was to investigate the nature of the model's errors. 12 out of 16 incorrect predictions were accompanied by clear explanations that enabled the experts to provide insightful feedback. These explanations fall into two categories: reasoning deficiencies (4) and legal knowledge deficiencies (8). Notably, the model demonstrated confusion regarding the distinction between federal and state courts in two instances, which is a critical aspect of legal knowledge. Table 9 presents an example in which the model appeared to comprehend the relevant legal knowledge, but ultimately failed due to constraints in its reasoning abilities.

6 Conclusion

This paper described the Archimedes-AUEB submission to SemEval-2024 Task 5. The submitted method is based on a LLM fine-tuned on CoT data in order to offer justifications for the predictions it generates. The training explanations are generated by GPT-3.5 in conjunction with expert analyses. Concurrently, we implement an innovative prompting procedure, dubbed Multiple Choice Mutation, designed to produce artificial instances. Both approaches contribute to a significant increase in performance. Our solution ranks 15th out of 20 contestants. Unlike other competitors, it generates explanations for its predictions. It also surpasses a strong GPT-3.5 baseline that can also produce explanations. We also assessed the output of our system qualitatively, with assistance from legal experts. In many cases the system generated high-quality explanations. The errors made by the model can be attributed to either its reasoning limitations or its deficiencies in legal knowledge. We recommend including similar qualitative evaluations in future work, as they reveal pivotal insights about the model that performance metrics alone cannot capture.

Figure 2: Correct predictions annotated for correct alignment with expert analysis.

Figure 3: Incorrect predictions annotated for clarity.

Correct prediction but inaccurate explanation
Question: A computer programmer living in Texas did some work for a corporation which is located in the Northern District of Illinois, but they refused to pay him. **Correct answer:** The court would have the authority to transfer to the Northern District of Illinois, but probably would not. **Expert analysis:** It is unlikely that the convenience of witnesses and the interests of justice require transfer of this case to Illinois. **Model output:** It would not be an appropriate venue, as the case should have been filed in the Texas district.

Table 8: Example where the model predicted the correct label, but failed to produce an accurate explanation due to a deficiency in legal knowledge.

Incorrect prediction but plausible explanation
Question: A sues his former corporation, claiming that he was fired based on his age. B, one of his former supervisors, is about to move to South Korea. A's counsel should **Correct answer:** prepare for the deposition, and cross-examine B as if he were doing so at trial. **Model output:** Incorrect, because a plaintiff who is planning to call B as a witness should not cross-examine her during the deposition to preserve her trial opportunities. **Legal expert feedback:** The AI failed to recognize that the witness planned to leave the country and there will not be any opportunity to examine him at trial.

Table 9: Example where the model understood legal concepts, but failed due to reasoning limitations.

Acknowledgements

We are most grateful to the legal experts, Philippe Jougleux, Associate Professor, School of Law, European University Cyprus, and Helen Bougas, J.D. Candidate, Southern Methodist University.

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the Next Generation Program.

References

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241, Online. Association for Computational Linguistics.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Joseph W Glannon. 2018. *The Glannon Guide To Civil Procedure: Learning Civil Procedure Through Multiple-Choice Questions and Analysis*, 4th edition. Aspen Publishing.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. Reinforced self-training (rest) for language modeling.

Lena Held and Ivan Habernal. 2024. SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer llama technical report. *ArXiv*, abs/2305.15062.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas

Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

A Appendix

A.1 Complete prompt examples

The following are examples with the complete prompts used for data augmentation with GPT-3.5 (see Table 10 for HGE, Table 11 for MCM-A and Table 12 for MCM-B). For each example we provide system instructions, an example of the task and the ideal response. During generation with GPT-3.5, we would also provide the actual training instance formatted as in the task example (Introduction, Question, Answer). To find the best prompts that worked well consistently we performed some initial experiments and inspected the quality of individual examples. We present examples of this process for Stage A of MCM in Appendix A.2.

Human-guided explanations
System instructions You are an expert lawyer in the domain of U.S. civil procedure. You are given an introduction to a legal case, a question, an answer and an analysis that justifies the correctness of that answer. These are taken from a legal textbook, however they are not formal and they refer to multiple choice letters (A, B, C, D). Read the given analysis carefully and explain why the answer is correct in a short, formal, clear way, without mentioning anything about letters A, B, C, D. **Task example** Introduction: Section 1391(d), which defines the residence of a corporation that has contacts in one district within a state but not others, is confusing. [...] Question: Arthur wishes to bring a diversity action in federal court against Cleveland Manufacturing Company. Cleveland has its factory and principal place of business in the Northern District of Illinois, but no other contacts with Illinois. The claim is based on alleged negligence in making a toaster at the Illinois factory, which caused a fire in Arthur's home in the Middle District of Georgia. Answer: The Southern District of Illinois is not a proper venue under §1391 because no events giving rise to the claim took place there and Cleveland does not reside there under the venue statute. Analysis: So, C is the right answer. The Southern District of Illinois is not a proper venue. **Ideal response** Explanation: The Southern District of Illinois cannot serve as a proper venue for Arthur's lawsuit against Cleveland Manufacturing Company under Section 1391(d) because Cleveland lacks sufficient contacts in this district to be considered as 'residing' there within the venue statute.

Table 10: Prompt instructions and one-shot example for generating human-guided explanations. The query for the training instance is not shown, but it follows the same structure as the 'Task example'.

Multiple choice mutation - Stage A

System instructions
You are an expert lawyer in the domain of U.S. civil procedure. You are given an introduction that describes a legal case, a question about that case and an answer to that question. First, you must identify what legal concept the question is talking about. Note that the given question contains some background on the details of the case. Provide a different background for your question and then ask a multiple choice question in the style of the original question, to illustrate that concept to law students.
Training instance
Introduction:
...
Question:
Chu, a Californian, went skiing at Aspen, in Colorado, which has only one federal judicial district. After he discovered that he was somewhat out of shape, he bought an exercise machine from Jackson, a friend from Kansas with a two-week time share in Aspen, who had brought the machine with him to Aspen after buying it in Texas. The machine was made by Sweda- Trak, a Texas company that sells its products only in Texas, and with its only place of business in the Western District of Texas. Chu brought the machine back to Los Angeles, California, where he was injured using it. Chu brings a diversity action against Jackson and Sweda-Trak in federal court, alleging that Jackson was negligent in maintaining the machine, and Sweda-Trak in designing it. Venue in Chu's action would be
Answer:
proper in the Southern District of California under §1391(b)(2), because a substantial part of the events giving rise to the claim took place there.
Ideal response (Mutated instance)
Concept:
Venue in Civil Procedure
Background:
Sophia, a New York resident, wants to sue Eric, a Texas resident, claiming that he infringed on her copyright by using her work in his recent publication. Sophia has never been to Texas, but she knows that Eric frequently visits New York for business meetings. According to U.S. law, venue refers to the most appropriate location for a trial. It can be the district where the defendant resides or where the events related to the claim occurred.
Question:
In which district would venue be proper if Sophia brings a copyright infringement lawsuit against Eric in federal court?
Choices:
A) The Southern District of New York, where Sophia resides
B) The Western District of Texas, where Eric resides
C) The Eastern District of New York, where Eric frequently visits for business meetings
D) The Northern District of Texas, where Sophia's work was originally created
Correct Answer:
A) The Southern District of New York, where Sophia resides

Table 11: Prompt instructions and one-shot example for generating a mutated question and candidate answers (choices/options) from an original training instance with MCM - Stage A.

Multiple choice mutation - Stage B

System instructions
You are an expert lawyer in the domain of U.S. civil procedure. You are given an introduction that describes legal concepts, one of these concepts, background of a legal case and a multiple choice question about that case. First, answer which answer is the correct one and then for each choice explain in one sentence why it is correct or wrong.
Task example
Introduction:
...
Concept:
Venue in Civil Procedure
Background:
Mark, a resident of Florida, wants to file a personal injury lawsuit against GreenCo, a multinational corporation headquartered in Delaware, for injuries sustained due to a defective product. The product was purchased online and delivered to Mark's home in Florida. Mark believes the defective product was manufactured and designed at GreenCo's facility in Kansas.
Question:
In which district would venue be proper if Mark brings a personal injury lawsuit against GreenCo in federal court?
Choices:
A) The District of Delaware, where GreenCo is headquartered
B) The Middle District of Florida, where Mark resides
C) The District of Kansas, where the defective product was manufactured and designed
D) The Southern District of Florida, where the product was delivered to Mark's home
Ideal response
The correct answer is:
B) The Middle District of Florida, where Mark resides
Choices:
A) The District of Delaware, where GreenCo is headquartered
Incorrect, because while GreenCo's headquarters are in Delaware, there doesn't seem to be any significant actions related to the claim that occurred there.
B) The Middle District of Florida, where Mark resides
Correct, because the product was delivered to Mark's home in Florida, causing the injury. Therefore, a significant portion of the events related to the claim occurred there.
C) The District of Kansas, where the defective product was manufactured and designed
Incorrect, although the product was manufactured and designed in Kansas, the significant harm occurred in Florida, making this district less relevant for this case.
D) The Southern District of Florida, where the product was delivered to Mark's home
Incorrect, because Mark does not reside in the Southern District of Florida and no specific information was given that suggest the Southern District as the proper venue.

Table 12: Prompt instructions and one-shot example for filtering inconsistent responses and generating explanations for each candidate answer from a mutated question and its candidate answers with MCM - Stage B.

A.2 Prompt engineering artifacts for MCM

As explained in Section 3.4, the initial prompt we came up with was not generating consistent responses from GPT-3.5 and for this reason we added the artifacts of 'concept' and 'question background'. We experimented with many different prompts and inspected their outputs to optimize these artifacts. Here we present an example of a training instance in Table 13 (it is the same example as in Table 1) and show the different mutated instances that would be produced with and without artifacts in Table 15 and Table 14 respectively. It is clear even from just one example that the response is much closer to the original training instance when we use artifacts. It is notable that we did not encounter the same problems with GPT-4.

Original training instance
Introduction: [...] the Supreme Court addressed this issue in Klaxon Co. v. Stentor Mfg. Co., 313 U.S. 487 (1941). Klaxon held that a federal diversity court should use the choice-of-law rules of the state in which it sits. ... Question: Your law or mine? Whitney, a Rhode Island citizen, goes skiing at a ski area in Vermont. When he purchases his lift ticket, he quickly signs a release of liability, waiving his right to sue, should he be injured due to negligence of the operator. While coming down an expert trail, he comes suddenly upon a bare spot full of rocks. He falls and is injured. He brings a diversity action against the operator in federal court in Rhode Island, basing personal jurisdiction on the fact that it had sent him an offer there for a weekend ski package that induced him to go to Vermont to ski. Assume that Rhode Island tort law refuses to enforce advance waivers of liability for negligence, unless they were individually negotiated. Assume further that Vermont tort law holds such waivers enforceable. Assume further that Rhode Island's choice-of-law rule for tort cases is to apply the law of the place of the injury, while Vermont's choice-of-law rule involves an assessment of the interests of the parties and states involved. It is less predictable under this balancing approach to choice of law whether the court would apply Rhode Island or Vermont law. The Rhode Island federal district judge would probably apply Answer: Vermont law to Whitney's case, because the accident happened in Vermont.

Table 13: Original training instance example.

Mutated question without artifacts
Vanilla prompt
You are an expert lawyer in the domain of U.S. civil procedure. You are given an introduction that describes a legal case, a question about that case and an answer to that question. Ask a multiple choice question in the style of the original question.
Response
Question: In a scenario where Emma, a Florida resident, sues Jack, a Georgia resident, for breach of contract in a federal court located in Georgia, which state's law would the federal court likely apply to the case under the principle established in Klaxon Co. v. Stentor Mfg. Co.? Choices: A) Florida law, where Emma resides B) Georgia law, where Jack resides C) Texas law, where the contract was signed D) New York law, where the federal court is located Correct Answer: B) Georgia law, where Jack resides

Table 14: The mutated question is missing some key elements when artifacts are not used, because GPT-3.5 has limited ability to understand complex prompts.

Mutated question with artifacts
Prompt with artifacts
You are an expert lawyer in the domain of U.S. civil procedure. You are given an introduction that describes a legal case, a question about that case and an answer to that question. **First, you must identify what legal concept the question is talking about. Note that the given question contains some background on the details of the case. Provide a different background for your question** and then ask a multiple choice question in the style of the original question, **to illustrate that concept to law students.**
Response
Concept: Choice of Law in Diversity Cases Background: Megan, a Florida resident, enters into a contract with Alex, a Georgia resident, for the purchase of a rare antique clock. The contract was negotiated over the phone and through emails, with the clock to be delivered to Megan in Florida. However, when the clock arrives, Megan discovers that it is damaged beyond repair. She decides to sue Alex in federal court in Georgia for breach of contract. The issue arises as to which state's law should govern the contract - Florida where Megan resides, or Georgia where Alex resides. Question: In a diversity action where Megan brings a breach of contract lawsuit against Alex, which state's law would a Georgia federal court likely apply, based on traditional choice-of-law doctrine? Choices: A) Florida law, where Megan resides B) Georgia law, where Alex resides C) The law of the state where the contract was formed, based on the traditional choice-of-law rule D) The law of the state with the most significant contacts to the contract, based on the traditional choice-of-law rule Correct Answer: C) The law of the state where the contract was formed, based on the traditional choice-of-law rule

Table 15: The mutated question is closer to our desired format when we use artifacts.

A.3 System response samples

In this section we present samples of complete instance-response pairs from our system (from the extended development set) annotated by legal experts. We present two samples of alignment between expert analysis and model explanation (Tables 16 and 17) and one sample that the explanation was not aligned (Table 18). We also present one sample where the prediction was wrong and the explanation was inacurrate because of legal deficiencies (Table 19), another one that failed because of limitations in reasoning cabilities (Table 20) and a sample that failed but the explanation was unclear to the legal experts (Table 21).

Aligned explanation - 1

Introduction:
The message of the Erie case is clear: "federal court: apply state law in a diversity case." [...] Every state has "choice-of-law" rules to deal with this problem, which arises frequently in state cases as well as federal diversity cases. [...] Three years after Erie, the Supreme Court addressed this issue in Klaxon Co. v. Stentor Mfg. Co., 313 U.S. 487 (1941). Klaxon held that a federal diversity court should use the choice-of-law rules of the state in which it sits. Thus, a New York federal court should apply whatever state substantive law the New York state court would apply to the case. If a New York court, under its choice-of-law rules, would apply New York tort law to Tompkins's accident (even though it took place in Pennsylvania), the New York federal court should do so as well. On the other hand, if a New York court would, under its choice-of-law rule for torts, apply Pennsylvania law to Tompkins's case, the New York federal court should similarly look to Pennsylvania law.
...
Question:
Your law or mine? Whitney, a Rhode Island citizen, goes skiing at a ski area in Vermont. When he purchases his lift ticket, he quickly signs a release of liability, waiving his right to sue, should he be injured due to negligence of the operator. While coming down an expert trail, he comes suddenly upon a bare spot full of rocks. He falls and is injured. He brings a diversity action against the operator in federal court in Rhode Island, basing personal jurisdiction on the fact that it had sent him an offer there for a weekend ski package that induced him to go to Vermont to ski. Assume that Rhode Island tort law refuses to enforce advance waivers of liability for negligence, unless they were individually negotiated. Assume further that Vermont tort law holds such waivers enforceable. Assume further that Rhode Island's choice-of-law rule for tort cases is to apply the law of the place of the injury, while Vermont's choice-of-law rule involves an assessment of the interests of the parties and states involved. It is less predictable under this balancing approach to choice of law whether the court would apply Rhode Island or Vermont law. The Rhode Island federal district judge would probably apply
Answer: Vermont law to Whitney's case, because the accident happened in Vermont.
Expert analysis:
Label: *Correct*
Although Rhode Island wouldn't enforce this waiver in a domestic case, its choice-of-law rule calls for application of the law of the place of injury, and that means Vermont law rather than its own.
Model explanation:
Label: *Correct*
The Rhode Island federal district judge would probably apply Vermont law to Whitney's case, as per Klaxon holding that the judge should use the local state's choice-of-law rules, resulting in Vermont law being likely applied.

Table 16: An example of alignment between expert analysis and model explanation (this is the complete prompt and response for the instance that was shown in Table 1).

Aligned explanation - 2

Introduction:
Rule 8 contains simple, sensible provisions governing the defendant's responses to the allegations in the complaint. First, Rule 8(b) requires the defendant to admit or deny each of the allegations in the complaint.
[...]
Under Rule 8(b)(2), a "denial must fairly respond to the substance of the allegation." If the defendant means to deny only a part of an allegation, she must specify what is true and deny only the parts that are not. Failing to deny an allegation constitutes an admission of it. Fed. R. Civ. P. 8(b)(6). The idea is that, if the parties plead conscientiously by the Rules, the parties and the court can compare the allegations in the complaint and the responses in the answer, and determine which issues are contested and which are agreed upon (and therefore need not be litigated).
Question:
Patterson sues Applied Electronics, Inc., for injuries suffered in an accident with Adair, one of AE's delivery drivers. Patterson alleges in Paragraph 4 of his complaint that Adair was an employee of AE at the time of the accident, that Adair was acting in the scope of employment at the time of the accident (which Patterson must prove in order to recover from the employer), that Adair was talking on his cell phone at the time of the accident, and that Adair's negligent driving caused the accident. AE answers Paragraph 4 as follows: "The defendant admits that Adair was its employee at the time of the accident, denies that Adair was negligent, but is without sufficient information to form a belief as to the truth of the allegation that Adair was talking on his cell phone at the time of the accident." Under Federal Rule 8,
Answer:
AE has admitted that Adair was its employee, but not that he acted in the scope of employment. The allegation that Adair was talking on his cell phone is treated as denied.
Expert analysis:
Label: *Incorrect*
This question simply requires you to compare Patterson's allegations with the defendant's responses, and apply the provisions of Rule 8 as to the effect of a denial, failure to deny, or allegation that the pleader is "without sufficient facts to form a belief." Naturally, AE has admitted that Adair was its employee, by admitting it! (Sometimes even the pleading rules are commonsensical.) The allegation that Adair was talking on his cell phone is treated as denied: Under the Rule 8(b)(5), the "without sufficient information" response is treated as a denial. Adair hasn't responded at all to the allegation that Adair acted in the scope of employment at the time of the accident. Frequently, defendants would like to ignore allegations in the complaint, if they aren't sure of their truth, or it would be awkward to admit them. But this head-in-the-sand strategy doesn't work under the Rules: Rule 8(b)(6) provides that allegations that are not denied are deemed admitted. So, effectively, AE has admitted that Adair was its employee and that he acted in the scope of employment, but denied negligence and that Adair was chatting on the cell phone.
Model explanation:
Label: *Incorrect*
The answer is incorrect because the language used by AE is not consistent with the treatment described in Rule 8(b)(6), which requires the defendant to deny only a part of the accusation if not admitting the whole.

Table 17: Another example of alignment between expert analysis and model explanation. Note that the model's explanation is much more concise.

Failed alignment of explanation

Introduction:
Sometimes a case is filed in a court that has subject matter jurisdiction over the case, personal jurisdiction over the defendant, and is a proper venue under the venue statutes, yet the court concludes, as a matter of common sense, that it should be litigated somewhere else.
[...]
In response to the Gulf Oil decision, Congress enacted 28 U.S.C. §1404(a), the federal transfer-of-venue statute. Under this section, a judge may transfer a case filed in her court that should be litigated elsewhere to the more appropriate federal district, instead of dismissing it for forum non conveniens. (a) For the convenience of parties and witnesses, in the interest of justice, a district court may transfer any civil action to any other district or division where it might have been brought or to any district or division to which all parties have consented. This statute makes good administrative sense. Because the federal district courts are all units of the same system, Congress has the authority to provide for transfer of cases among them to promote efficiency.
...
Question:
The federal shuffle. Connors, a computer programmer living in the Northern District of Texas, did some contract work for Ace Corporation, which is incorporated in Delaware with its principal place of business in the Northern District of Illinois. Ace has an office in the Northern District of Texas, but no other offices in Texas. It has large operations in three other states. Connors negotiated the contract by phone and e-mail from his home, dealing with Ace employees in the Texas office and the Illinois office. He did the programming work, which was to be used for Ace's Texas operation, at his home. Ace Corporation claimed that Connors had only done half the work, and refused to pay. Connors, believing that he had completed all the work called for under the contract, brought a diversity action against Ace for the contract price ($200,000) in federal court in the Northern District of Texas.
Answer:
The court would have the authority to transfer to the Northern District of Illinois, but probably would not.
Expert analysis:
Although the court would have authority under 28 U.S.C. §1404(a) to transfer this case to the Northern District of Illinois, why would it? Most of the events giving rise to the claim took place in the Northern District of Texas. Connors did the work there, dealt with Ace employees there, and suffered the damages there. The programming was to be used in Ace's Texas office. The defendant, a corporation with a local office, would not be greatly inconvenienced or put to great expense to litigate this case in Texas. Some witnesses and evidence may be in Illinois, but it seems very doubtful that more would be there than in Texas. It seems highly unlikely that the convenience of witnesses and the interests of justice require transfer of this case to Illinois.
Model explanation:
The federal court in Texas would have the authority to transfer the case to the Northern District of Illinois under §1404(a), but the court would probably not do so because it would not be an appropriate venue, as the case should have been filed in the Texas district.

Table 18: A sample that the model explanation failed to align with the expert's analysis (this is the complete prompt and response for the instance that was shown in Table 8).

Incorrect response due to legal deficiency

Introduction:
In Hanna, the Court addressed the power of Congress to promulgate rules that govern procedure in the federal courts. See 380 U.S. at 464. Analytically, it had to do so, because the power to write the Federal Rules of Civil Procedure is granted by Congress to the Court, in the Rules Enabling Act. Surely, Congress could not delegate to the Court any broader rule-making power than it possesses itself. In Hanna, the Court describes Congress's rule-making authority as follows: the constitutional provision for a federal court system (augmented by the Necessary and Proper Clause) carries with it congressional power to make rules governing the practice and pleading in those courts, which in turn includes a power to regulate matters which, though falling within the uncertain area between substance and procedure, are rationally capable of classification as either. 380 U.S. at 472. That's a mighty broad test for Congressional power over federal court procedure: If the provision can rationally be viewed as regulating procedure, Congress may enact it, even if it will affect the outcome of litigation. Consider, however, how the Court would rule if Congress enacted a statute that impinges on state law rights that can reasonably be classified as "substantive."
Question:
Limits on limitations. Congress, in an effort to avoid complex and time-consuming litigation on the proper time for filing diversity cases in the federal courts, enacts the Federal Diversity Limitations Act (FDLA), establishing a two-year limitations period for diversity cases in the federal courts, regardless of the applicable state limitations period for the claim. Sanchez brings a diversity action against Bernstein in federal court for fraud arising out of a real estate transaction. The relevant state limitations period for fraud claims is four years. Sanchez's suit is brought two years and five months after the alleged fraud was committed. The federal court should
Answer:
dismiss the claim, because the FDLA applies and bars it.
Expert analysis:
While this limitations period would not apply if it were a matter of federal judicial practice (see York), and probably could not be validly adopted as a Federal Rule (see 28 U.S.C. §2072(b)), there's a strong argument, under Hanna's description of congressional power over procedure, that this statute could be enacted by Congress. After all, under Hanna, Congress has very broad authority to regulate procedure in the federal courts. You could imagine that Congress might seek to protect those courts from expending limited judicial resources on state claims by imposing a limitations period on diversity cases. It is very likely within its power to do so, and should be. Of course, it hasn't, and probably won't pass such a limitations statute, since such a statute would interfere significantly with the administration of state causes of action. The members of Congress do, after all, represent the states.
Model explanation:
Label: *Incorrect*
The answer is incorrect because federal courts should not dismiss a case on the basis of federal statutes of limitations if state law permits the claim to be raised, unless there are grounds not based on the applicable federal limitations period to question admissibility of the claim.
Legal expert feedback:
The system has difficulty distinguishing the differences between state and federal court.

Table 19: An example where the model demonstrated confusion in distinguishing the state with the federal court.

Incorrect response due to limited reasoning capabilities

Introduction:
A deposition is the examination of a witness under oath. Depositions are the most effective means of previewing the detailed testimony of witnesses.
[...]
Depositions serve several purposes. They "get the witness on the record." That is, the witness tells her story under oath. If she contradicts that story at trial, she can be impeached with her deposition testimony.
[...]
However, in some cases a party may depose a witness in order to obtain the party's testimony for use at trial. If, for example, a witness is moving out of the country, the party who wants to use that witness's testimony at trial may take her deposition as a substitute for her live testimony at the trial. The deposition will be admissible at trial if the witness is not available to testify in person. Fed. R. Civ. P. 32(a)(4). Note that in this scenario, unlike the typical deposition, the party noticing the deposition will be the party who plans to use that witness's testimony, not the opposing party.
Question:
Your witness.
Czonka is fired from his job at fifty-nine, after twenty-one years with NorthStar Corporation. He sues in federal court under the Federal Age Discrimination in Employment Act (ADEA), claiming that he was fired based on his age. Two months after filing suit, Czonka's counsel learns that Roh, one of his supervisors at NorthStar, has just retired, and is about to move to South Korea. Although discovery has just begun, NorthStar has noticed Roh's deposition.
Czonka's counsel should
Answer:
prepare as fully as possible for the deposition, and cross-examine Roh as if she were doing so at trial.
Expert analysis:
Label: *Correct*
If Roh does not appear at trial, cross-examining at the deposition will be the only opportunity Czonka's counsel has to cross- examine Roh under oath. She had better do it as well as she can.
Model explanation:
Label: *Incorrect*
The answer is incorrect because a well-prepared plaintiff who is planning to call Roh as a witness should not cross-examine her during the deposition to preserve her trial opportunities, but should focus on clarifying points and getting her story on the record.
Legal expert feedback:
AI based the reasoning on trial opportunity whereas the protagonist planned to leave the country and there will not be any opportunity to examine him at trial.

Table 20: An example where the model failed, even though it understood legal knowledge, due to limited reasoning capabilities (this is the complete prompt and response for the instance that was shown in Table 9).

Incorrect response with unclear explanation

Introduction:
The discussion so far has focused on motions by the defendant. But the plaintiff may seek summary judgment as well, under the same standard: that there is no genuine issue of material fact and she is entitled to judgment as a matter of law. However, the effect of summary judgment for the plaintiff is often different from the grant of the motion for a defendant.
Question:
Insulation from liability. Maxwell sues Chao for breach of a contract to insulate Chao's house. He alleges that they had a contract, and that he did the work, but Chao refused to pay the agreed price for the work. Maxwell seeks recovery in the alternative on a quantum meruit theory. Chao denies that Maxwell can sue for breach of contract, because the writing Maxwell relies on as the contract—a scribbled note indicating the price for the job—fails to satisfy the requirements for a valid contract under the relevant statute of frauds. Chao also claims that the insulation used was substandard and improperly installed. Maxwell moves for summary judgment on the question of whether the parties had a contract. In support of his motion, he submits the scribbled note itself, an affidavit attesting that it was signed by Chao in his presence, and a brief arguing that the note is sufficient to constitute a binding contract under the statute of frauds. Chao submits no opposing materials, but submits a brief arguing that the note does not constitute a contract, because the terms are insufficiently described in it. **Answer:**
If the judge concludes that the note meets the requirements of a valid contract, she should enter partial summary judgment for Maxwell on the issue of the validity of the written contract.
Expert analysis:
Label: *Correct*
If the judge concludes that the note suffices as a writing, she should grant partial summary judgment for Maxwell on the issue of the sufficiency of the note as a contract. But she would not enter judgment for Maxwell on the claim as a whole, because his motion only establishes one element of the claim, existence of the contract. Maxwell still has to establish other elements of the contract claim (such as adequate performance) or the quantum meruit claim, in order to recover from Chao.
Model explanation:
Label: *Incorrect*
The answer is incorrect because while a judge may enter partial summary judgment on the issue of the written contract's validity, the plaintiff's alternative claim is the subject of different issues, and partial summary judgment will only cover those issues where there is no genuine issue of material fact.

Table 21: An example of an incorrect explanation that the experts found unclear.

A.4 Experts' evaluation

For the manual evaluation process by legal experts we created Google Forms automatically with the Google Apps Script software.[3] Each form contains eight examples in total, four of which are correct labels and the other four incorrect. The experts were given general instructions (Table 22) in order to fill in the forms as consistently as possible. They were also given specific instructions for each task (Table 23). Each example in the form contained an input instance(introduction, question, answer and label of answer, expert analysis) and the prediction of Llama-2-MCM along with the generated explanation for that particular instance.

General Instructions for experts
Outlined Steps Carefully review the Question, Answer, and corresponding Labels. Read the Human Explanation briefly to gain a preliminary understanding. Read the AI explanation and evaluation question carefully. If you know the evaluation question, provide a direct response. Otherwise, revisit the skipped sections for further clarification. In cases of uncertainty after revisiting the essential sections, you can proceed to the subsequent question. The objective is to provide definitive answers wherever possible while optimizing time efficiency. Note that sometimes the explanations (either human or AI), refer to letters, such as "A is correct". Please ignore these, as the letters used to refer to multiple choice questions, but they have been shuffled. Disclaimer: All legal scenarios presented are fictitious and derived from 'The Glannon Guide To Civil Procedure: Learning Civil Procedure Through Multiple-Choice Questions and Analysis, 4th edition.'

Table 22: General instructions provided to legal experts before they fill in the Google Form.

Correct Predictions
In this scenario, the AI has correctly predicted an outcome, and we aim to assess whether it did so for the right reasons or merely by chance. To accomplish this, carefully read both the Question and the Answer, followed by the Human explanation. Next, review the AI explanation and determine if it seems plausible and consistent with the human explanation. Consider whether the AI truly comprehended the reasoning behind the answer. Please refrain from comparing it to the Human explanation, as our intention is not to ascertain whether the AI outperforms the human (which it does not, of course). If you're unsure about what the AI actually understood, you may leave your answer blank, but we encourage you to respond to as many samples as possible.
Incorrect Predictions
In this scenario, the AI has provided an incorrect prediction, resulting in an erroneous explanation. However, our objective is to assess the frequency with which the AI engages in what is known as "hallucination", where it imagines an explanation without genuine reasoning, versus instances where it genuinely attempts to reason but arrives at an incorrect conclusion. In the former case, akin to a student failing to put effort into their homework, we aim to identify instances where the AI requires correction. If you find the purpose of the AI explanation unclear, please indicate as such.

Table 23: Instructions for the alignment task (Correct) and the task about the clarity of incorrect predictions (Incorrect) provided to legal experts before they fill in the Google Form.

[3] https://www.google.com/script/start/

SemEval-2024 Task 8: Weighted Layer Averaging RoBERTa for Black-Box Machine-Generated Text Detection

Ayan Datta[†]
IIIT Hyderabad
ayan.datta
@research.iiit.ac.in

Aryan Chandramania[†]
IIIT Hyderabad
aryan.chandramania
@research.iiit.ac.in

Radhika Mamidi
IIIT Hyderabad
radhika.mamidi@iiit.ac.in

Abstract

This document contains the details of the authors' submission to the proceedings of SemEval 2024's Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection Subtask A (monolingual) and B. Detection of machine-generated text is becoming an increasingly important task, with the advent of large language models (LLMs). In this paper, we lay out how using weighted averages of RoBERTa layers lets us capture information about text that is relevant to machine-generated text detection.

1 Introduction

Language modeling is a foundational task in NLP, and encompasses learning of all the features that make up language. Different levels of linguistic information are stored in language models' (LM) hidden states. This may include syntax, morphological features, phrasing, and so on. (Rogers et al., 2021) Our aim is to leverage this encoded information to help us discern machine-generated text.

The advent of large language models (LLMs) has transformed the digital landscape, and this has also led to the proliferation of machine-generated text in spaces spanning from legal proceedings, to articles, to school submissions. With this, there has been a consequential rise in the need to be able to distinguish between machine- and human-generated text across domains. Just as important is the need to be able to identify the generators for text that has been flagged as being generated by machines.

In this paper, we describe our methodology and attempts to create a system that can perform the task effectively.

2 System Overview

We have used RoBERTa-base for all experiments in the scope of this paper. The baseline set by the task organizers is reported to have been from a finetuned RoBERTa model. RoBERTa has the same architecture as BERT, but uses a byte-level BPE as a tokenizer and uses a different pretraining scheme and has become a SOTA model since its release (Liu et al., 2019).

2.1 Weighted Layer Averaging

The standard fine-tuning setup uses the [CLS] Representation of the last layer of RoBERTa. It has been shown that different layers of BERT-like models capture different levels of linguistic information, the lower layers capture lexical information and word order,the middle layers capture syntactic information, and the higher layers capture semantic and task specific information (Rogers et al., 2020). We believe that using just the last layer representation may discard some of the syntactic and lexical information, which could be crucial for the task of detecting machine generated text. We use the weighted sum of all the token representations, where each layer is assigned a corresponding weight, trained along with the downstream task, similar to ElMo (Peters et al., 2018). Let $x_0, x_1, ...x_n$ be the input sequence. Roberta generates the following hidden states.

$$\texttt{RoBERTa}([x_0, x_1, \ldots, x_n]) = H$$

Where H is a matrix consisting of hidden state vectors $\mathbf{h}_i^j$ corresponding to the j^{th} layer, and the i^{th} token. $i = 0$ represents the embedding layer

[†]These authors contributed equally to this work.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1623–1626

output.

$$H = \begin{bmatrix} \mathbf{h}_0^0 & \mathbf{h}_1^0 & \dots & \mathbf{h}_n^0 \\ \mathbf{h}_0^1 & \mathbf{h}_1^1 & \dots & \mathbf{h}_n^1 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_0^{12} & \mathbf{h}_1^{12} & \dots & \mathbf{h}_n^{12} \end{bmatrix}$$

The standard fine-tuning setup uses $\mathbf{h}_0^{12}$ which corresponds to the [CLS] token and passes it through another Feed Forward Network to get the output class probabilities. We propose averaging all of the layer hidden states. The input **y** to the Feed Forward Network that produces the class probabilities is computed as follows.

$$\mathbf{y} = \frac{1}{12} \cdot \sum_{j=0}^{12} \frac{\lambda_j \sum_{i=0}^{n} \mathbf{h}_i^j}{n}$$

λ_j is the layer weight assigned to the layer j. $[\lambda_0, \lambda_1, ...\lambda_{12}]$ are trained along with the classification task.

2.2 Parameter Efficient Tuning with AdaLoRa

A full continual finetune of RoBERTa (and LLMs, in general) with all the weights being updated is known to potentially lead to catastrophic forgetting (Ramasesh et al., 2022), which may cause the model to become unable to generalize, with the pretraining being, for all intents and purposes, in vain.

It has also been shown that common pre-trained models have a very low intrinsic dimension; in other words, there exists a low dimension reparameterization that is as effective for fine-tuning as the full parameter space (Aghajanyan et al., 2020). This implies that full continual finetuning – being potentially harmful as well as unnecessary – can be replaced with a better, more parameter efficient method, which grants us more freedom with regards to model and data sizes.

Low-rank Adapters (LoRA) (Hu et al., 2021) were designed with this in mind. LoRA freezes the pretrained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks. They also offer an improvement over unfreezing just the last few layers by attaching to every layer in the model, which allows them to modify information flow at every step, starting from the source.

For our task, we made use of Adaptive LoRA (AdaLoRA) (Zhang et al., 2023), which adjusts the matrices based on parameters learned during training, i.e. the ranks of the adapters themselves are learned. Our hope is that by doing this, we prevent unnecessarily large adapters where there is not much to do, and conversely provide the flexibility to have larger matrices to handle greater amounts of information change.

3 Data

Data for the task was provided by the organizers(Wang et al., 2024a). It is an extension of the M4 Dataset (Wang et al., 2024b). The name stands for multi-generator, multi-domain, and multi-lingual corpus for machine-generated text detection. As the name suggests, the dataset has been created with text from different generators spanning multiple domains. The data for subtask A and B follow the same format, consisting of source (such as Wikipedia), model (such as Dolly), label (such as Human), and the text to be classified. The data for subtask C contains text with a combination of human- and machine-generated text, and a label indicating the word index at which the split occurs.

For our experiments, we resplit the training and dev datasets and split them uniformly across generators and domains in an 80-20 split‘. Our split of the dev set is bigger than the official dev set, to get a better estimate of our model’s performance.

4 Experimental Setup

We use RoBERTa’s tokenizer and trained our models for Subtask A (monolingual) (Binary Classification) and Subtask B (Multi-Class Classification) on the resplit train data and use the resplit evaluation data for early stopping. Our Hyperparameter Configuration has been specified in Appendix A.

5 Results

Our model while doing really well on our evaluation set, falls short on the test set scoring around 13 percentage points lower than the baseline for subtask A and around 1 percentage point lower than the baseline for subtask B. This could be attributed to the model not being as good in generalising to unseen domains and generators. We hypothesize more hyperparameter tuning, better aggregation of

Model	Accuracy
Ours	0.7535
Baseline	0.8846

Table 1: Results for Subtask A as computed by the organizers

Model	Accuracy
Ours	0.7387
Baseline	0.7460

Table 2: Results for Subtask B as computed by the organizers

the token representations than averaging by utilizing models like LSTMs (Hochreiter and Schmidhuber, 1997), may help the model better generalise to unseen domains and generators by being able to capture more complex features and patterns. The submission scores as computed by the task organizers have been reported in Tables 1 and 2. Scores on our Validation, the official validation and the official test set as computed by us have been reported in Tables 3 and 4.

6 Conclusion

We have demonstrated that linguistic information encoded in the various layers of large language models such as RoBERTa can be used to effectively demonstrate if a text is machine-generated or not, across different domains and generators.

References

[Aghajanyan et al.2020] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Hu et al.2021] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

[Liu et al.2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

[Peters et al.2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

[Ramasesh et al.2022] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2022. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

[Rogers et al.2020] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works.

[Rogers et al.2021] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 01.

[Wang et al.2024a] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico, June.

[Wang et al.2024b] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multigenerator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta, March.

[Zhang et al.2023] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning.

A Hyperparameters

[†]The code used can be found in this repository: https://github.com/advin4603/AI-Detection-With-WLA

Dataset	Precision	Recall	Accuracy	F1 Score
Our Dev	0.9841	0.9949	0.9900	0.9895
Official Dev	0.9744	0.9444	0.9598	0.9592
Official Test	0.6823	0.9942	0.7538	0.8092

Table 3: Results for Subtask A as computed by us

Dataset	$\text{Precision}_{\text{Micro}}$	$\text{Recall}_{\text{Micro}}$	Accuracy	$\text{F1 Score}_{\text{Micro}}$
Our Dev	0.979	0.979	0.979	0.979
Official Dev	0.9783	0.9783	0.9783	0.9783
Official Test	0.7398	0.7398	0.7398	0.7398

Table 4: Results for Subtask B as computed by us

Hyperparameter	Value
Learning Rate	5e-4
Batch Size	8
Weight Decay	5e-5
Warmup Ratio	0.1
init_r	12
target_r	8
lora_alpha	200
lora_dropout	0.4

Table 5: Hyperparameters for Subtask A (Monolingual)

Hyperparameter	Value
Learning Rate	5e-4
Batch Size	8
Weight Decay	5e-5
Warmup Ratio	0.01
init_r	12
target_r	8
lora_alpha	200
lora_dropout	0.4

Table 6: Hyperparameters for Subtask B

Mast Kalandar at SemEval-2024 Task 8: On the Trail of Textual Origins: RoBERTa-BiLSTM Approach to Detect AI-Generated Text

Jainit Sushil Bafna
IIIT Hyderabad
jainit.bafna@research.iiit.ac.in

Hardik Mittal*
IIIT Hyderabad
hardik.mittal@research.iiit.ac.in

Suyash Sethia*
IIIT Hyderabad
suyash.sethia@research.iiit.ac.in

Manish Shrivastava
IIIT Hyderabad
m.shrivastava@iiit.ac.in

Radhika Mamidi
IIIT Hyderabad
radhika.mamidi@iiit.ac.in

Abstract

Large Language Models (LLMs) have showcased impressive abilities in generating fluent responses to diverse user queries. However, concerns regarding the potential misuse of such texts in journalism, educational, and academic contexts have surfaced. SemEval 2024 introduces the task of Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection, aiming to develop automated systems for identifying machine-generated text and detecting potential misuse. In this paper, we i) propose a RoBERTa-BiLSTM based classifier designed to classify text into two categories: AI-generated or human ii) conduct a comparative study of our model with baseline approaches to evaluate its effectiveness. This paper contributes to the advancement of automatic text detection systems in addressing the challenges posed by machine-generated text misuse. Our architecture ranked 46th on the official leaderboard with an accuracy of 80.83 among 125.

1 Introduction

The task of classifying text as either AI-generated or human-generated holds significant importance in the field of natural language processing (NLP). It addresses the growing need to distinguish between content created by artificial intelligence models and that generated by human authors, a distinction crucial for various applications such as content moderation, misinformation detection, and safeguarding against AI-generated malicious content. This task is outlined in the task overview paper by (Wang et al., 2024), emphasizing its relevance and scope in the NLP community.

Our system employs a hybrid approach combining deep learning techniques with feature engineering to tackle the classification task effectively. Specifically, we leverage a BiLSTM (Bidirectional Long Short-Term Memory) (Schuster and Paliwal, 1997) neural network in conjunction with RoBERTa (Liu et al., 2019), a pre-trained language representation model, to capture both sequential and contextual information from the input sentences. This hybrid architecture enables our system to effectively capture nuanced linguistic patterns and semantic cues for accurate classification.

Participating in this task provided valuable insights into the capabilities and limitations of our system. Quantitatively, our system achieved competitive results, ranking 46 relative to other teams in terms of accuracy and F1 score. Qualitatively, we observed that our system struggled with distinguishing between sentences generated by AI models trained on specific domains or datasets with highly similar linguistic patterns.

We have released the code for our system on GitHub[1], facilitating transparency and reproducibility in our approach.

2 Related Works

In the field of detecting machine-generated text, numerous methodologies and models have been examined. A distinguished methodology is the application of the RoBERTa Classifier, which enhances the RoBERTa language model through fine-tuning for the specific purpose of identifying machine-generated text. The proficiency of pre-trained classifiers like RoBERTa in this domain has been affirmed through various studies, including those conducted by (Solaiman et al., 2019) and additional research by (Zellers et al., 2019; Ippolito et al., 2019; Bakhtin et al., 2020; Jang et al., 2020; Uchendu et al., 2021). Concurrently, the XLM-R Classifier exploits the multilingual training of the XLM-RoBERTa model to effectively recognize machine-generated text in various languages, as demonstrated by (Conneau et al., 2019).

*Equal contribution.

[1] https://github.com/Mast-Kalandar/SemEval2024-task8

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1627–1633

a)

Model/Source	chatGPT	cohere	davinci	dolly	human
wikihow	3000	3000	3000	3000	15499
wikipedia	2995	2336	3000	2702	14497
reddit	3000	3000	3000	3000	15500
arxiv	3000	3000	2999	3000	15498
peerread	2344	2342	2344	2344	2357

b)

Model/Source	bloomz	human
wikihow	500	500
wikipedia	500	500
reddit	500	500
arxiv	500	500
peerread	500	500

Table 1: Table a) contains statistics about the train split. Table b) contains statistics about the validation split from the M4 dataset

Alternatively, the exploration of logistic regression models that incorporate GLTR (Giant Language model Test Room) features has been undertaken. These models strive to discern subtleties in text generation methodologies by analyzing token probabilities and distribution entropy, as investigated by (Gehrmann et al., 2019). Furthermore, detection efforts have utilized stylometric and NELA (News Landscape) features, which account for a broad spectrum of linguistic and structural characteristics, including syntactic, stylistic, affective, and moral dimensions, as reported by (Li et al., 2014) and (Mitchell et al., 2023). Additionally, proprietary frameworks like GPTZero, devised by Princeton University, focus on indicators such as perplexity and burstiness to analyze texts for machine-generated content identification. Although the specific technical details are sparingly disclosed, the reported effectiveness of GPTZero in identifying outputs from various AI language models highlights its significance in the ongoing development of machine-generated text detection strategies (Ouyang et al., 2022; Brown et al., 2020; Radford et al., 2019; Touvron et al., 2023).

3 Background

3.1 Dataset

For the machine-generated text, the researchers used various multilingual language models like ChatGPT(OpenAI, 2024), textdavinci-003(OpenAI, 2022), LLaMa(Touvron et al., 2023), FlanT5(Chung et al., 2022), Cohere(Cohere, 2024), Dolly-v2(databricks, 2022), and BLOOMz(Muennighoff et al., 2023). These models were given different tasks like writing Wikipedia articles, summarizing abstracts from arXiv, providing peer reviews, answering questions from Reddit and Baike/Web QA, and creating news briefs. As evident from Table 1, the training set lacks any sentences generated by the Bloomz model, which stands as the sole model represented in the validation set. This deliberate choice ensures a robust assessment of our model's generalization capabilities across all machine-generated outputs, regardless of the specific model generating them. By exposing our model to diverse machine-generated sentences during training, including those from unseen models like Bloomz in the validation set, we aim to evaluate its ability to effectively generalize to novel inputs and make reliable predictions across the spectrum of machine-generated text.

3.2 Task

We focused on Subtask-A of the SemEval Task 8 which involves developing a classifier to differentiate between monolingual sentences generated by artificial intelligence (AI) systems and those generated by humans. This classification task is essential for distinguishing the origin of text and understanding whether it was produced by AI models or by human authors.

3.2.1 Objective

The primary objective is to build a robust classifier capable of accurately distinguishing between AI-generated and human-generated sentences. The classifier should generalize well across various AI models and domains, ensuring consistent perfor-

Model	Accuracy	F1	Precision	Recall	Params*
Full RoBERTa fine tune	80.68	80.54	81.55	80.68	124M
LoRA with RoBERTa (Freezed)	81.59	81.06	85.64	**81.59**	0.7M
LoRA with LongFormer	75.34	75.14	76.16	75.34	6M
BiLSTM with RoBERTa (Un-Freezed)	70.77	61.15	**91.19**	46.00	18M
GRU with RoBERTa (Freezed)	74.65	80.54	81.55	80.68	3M
BiLSTM with RoBERTa (Freezed)	**82.52**	**82.14**	83.96	80.40	4M

Table 2: The performance of the models tried on the dev set of the dataset.
*The params only accounts for trainable unfreezed parameters.

mance regardless of the specific model or domain from which the text originates.

The goal was to design a model that not only performs this task with high accuracy but also adapts to various AI models and domains. It's crucial for the classifier to accurately identify the origin of sentences, regardless of the technology used to generate them or their subject matter, ensuring broad applicability and effectiveness

4 System Overview

Based on our observation (See 7), we discovered that language modeling task encodes the various features required for detection of AI written text. So we used pretrained RoBERTa in most of our architectures so exploit this power of language models.

4.1 Full RoBERTa Finetune

The Full RoBERTa(Liu et al., 2019) Finetune model, chosen as our baseline, boasted an extensive architecture and possessed the highest parameter count among the models under evaluation. Serving as a comprehensive starting point, this model allowed us to assess the effectiveness of subsequent enhancements in comparison.

4.2 LoRA with RoBERTa (Frozen)

Incorporating Low Rank Adapters (Hu et al., 2021), we applied fine-tuning techniques to the RoBERTa model while strategically freezing all layers. This approach enabled us to adapt the model to our specific task domain, leveraging pre-trained representations effectively.

4.3 LoRA with LongFormer

The limitation of RoBERTa's context length (max 512 tokens) posed challenges for handling lengthy sentences in our dataset. To address this, we investigated LongFormer (Beltagy et al., 2020), a model designed to efficiently manage longer contexts. Despite employing LoRA for fine-tuning, the model's performance on the validation set fell short of expectations, indicating potential difficulties in generalization.

4.4 RoBERTa (2 Layers unfreezed) + BiLSTM

Expanding upon RoBERTa's capabilities, we introduced a hybrid architecture by unfreezing two layers and integrating a BiLSTM network (Schuster and Paliwal, 1997). RoBERTa served as the primary encoder for sentence representations, with the subsequent BiLSTM layer trained to classify based on the last hidden state.

4.5 RoBERTa (Frozen) + GRU

In our endeavor to augment RoBERTa's capabilities, we devised a hybrid architecture by integrating a Gated Recurrent Unit (GRU) (Chung et al., 2014) network with the frozen RoBERTa model. Within this framework, RoBERTa served as the encoder for generating sentence representations, while a subsequent GRU layer was incorporated for sequential processing and classification tasks. This amalgamation aimed to leverage the strengths of both RoBERTa's contextual understanding and GRU's recurrent dynamics, contributing to enhanced performance on our target task.

4.6 RoBERTa (Frozen) + BiLSTM

In our pursuit of enhancing RoBERTa's capabilities, we devised a hybrid architecture by coupling a Bidirectional Long Short-Term Memory (BiLSTM) network with the RoBERTa model (Liu et al., 2019). In this setup, RoBERTa functioned as the encoder for sentence representations, while a subsequent BiLSTM layer was employed for classification, utilizing the last hidden state for decision-making. For a detailed visual representation of the model's architecture, please refer to the accompanying Figure

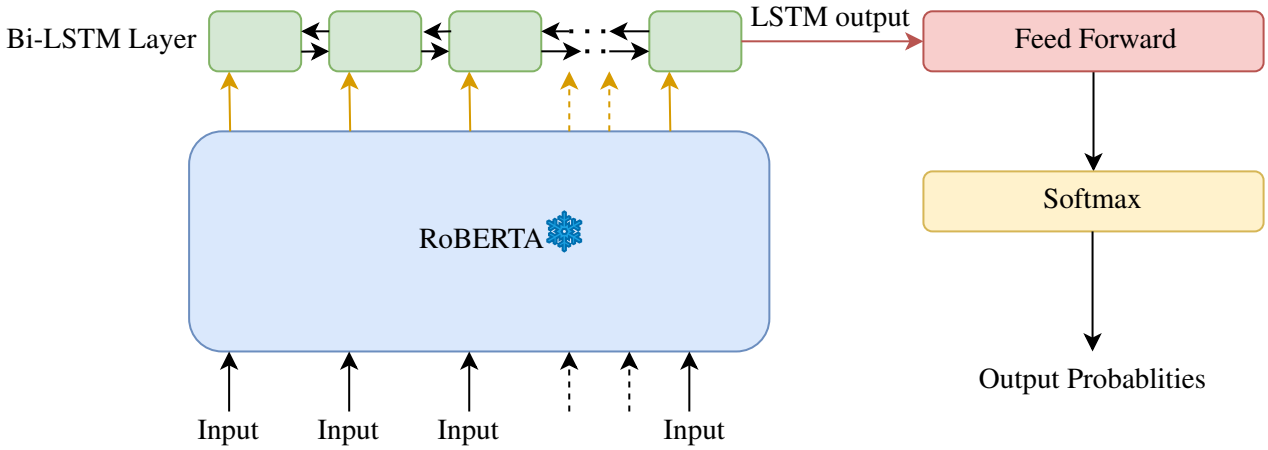


Figure 1: Our proposed architecture of BiLSTM with freezed RoBERTa

1.

We explored various methodologies (refer to Table 2 for detailed performance metrics) before selecting the optimal approach as our final model. Subsequently, we assessed the performance of the chosen model, RoBERTA (Freezed) + BiLSTM, on the test dataset.

5 Experiments

5.1 Preprocessing

All textual data underwent standard preprocessing steps, including tokenization, lowercasing, and punctuation marks. Additionally, specific domain-related preprocessing, such as handling special characters or domain-specific terms, was performed as necessary.

5.2 Hyperparameter Tuning

Hyperparameters were tuned using a combination of grid search and random search techniques. We explored various hyperparameter combinations to identify the optimal configuration for each model variant.

The configuration for LSTM and GRU used in Table 2 is `hidden_size=256`, `layers=2`, `dropout=0.2`, with `LoRA rank being 20` has been found as the best configuration for the models. For RoBERTa+LSTM model's feedforward had a single weight matrix of dimension 512*2.

6 Results

We tested our models on various models on the test set. The results can be viewed in (Table: 3).

Ranking: Our BiLSTM+RoBERTa model achieved a ranking of 46 out of 125 participants in the competition, demonstrating its competitive performance (as shown in Table 3). These results highlight the effectiveness of various models, including BiLSTM+RoBERTa and GRU+RoBERTa, in addressing the task objectives. We submitted BiLSTM+RoBERTa based on its strong performance on the validation set. However, after testing all models listed in Table 3, we found that GRU+RoBERTa achieved a significantly better result, with an accuracy increase of approximately 4%.

7 Conclusion

In conclusion, our BiLSTM+RoBERTa model effectively tackled the task, achieving competitive results, thanks to its deep learning and pre-trained language model. While a similar model with unfrozen RoBERTa boasted higher precision, its complexity came at the cost of increased parameters.

Impressively, our model ranked 46th out of 125 competition entries (Table 3), showcasing its potential alongside approaches like GRU+RoBERTa. Interestingly, post-competition analysis revealed GRU+RoBERTa's superior accuracy (by about 4%). This highlights the value of exploring diverse architectures and hyperparameter tuning for peak performance.

Moving forward, there are several avenues for future work to explore. Firstly, further experimentation with different model architectures, including alternative combinations of encoders and classifiers, could potentially yield improvements in performance. Additionally, fine-tuning hyperparameters and exploring advanced techniques for model optimization may enhance the robustness and generalization capabilities of our system. Furthermore,

Model	Accuracy	F1	Precision	Recall	Params*
Full RoBERTa fine tune^{+}	**88.47**	**88.44**	**93.36**	84.02	124M
LoRA with RoBERTa (Freezed)	80.91	80.18	83.88	80.14	0.7M
LoRA with LongFormer	63.39	57.51	72.45	61.67	6M
BiLSTM with RoBERTa (Un-Freezed)	80.80	80.19	83.08	80.12	18M
GRU with RoBERTa (Freezed)	84.71	84.33	86.53	84.13	3M
BiLSTM with RoBERTa (Freezed)	80.83	80.83	74.65	**96.16**	4M

Table 3: The performance of the models tried on the test set of the dataset.
* The params only accounts for trainable unfreezed parameters.
+ Baseline mentioned in task overview paper

incorporating additional contextual information or domain-specific knowledge could potentially augment the model's understanding and performance on specific tasks. Overall, our findings contribute to the ongoing research efforts in natural language processing and provide valuable insights for future developments in this domain.

References

Anton Bakhtin, Yuntian Deng, Sam Gross, Myle Ott, Marc'Aurelio Ranzato, and Arthur Szlam. 2020. Energy-based models for text. *CoRR*, abs/2004.10188.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling.

Cohere. 2024. Cohere: Chat. https://cohere.com/. Accessed: February 20, 2024.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

databricks. 2022. Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM. dolly-v2. Accessed: February 20, 2024.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Human and automatic detection of generated text. *CoRR*, abs/1911.00650.

Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, Sang-ug Kang, and Jong Wook Kim. 2020. Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism. *Applied Sciences*, 10(17).

Jenny S. Li, John V. Monaco, Li-Chiou Chen, and Charles C. Tappert. 2014. Authorship authentication using short messages from social networking sites. In *2014 IEEE 11th International Conference on e-Business Engineering*, pages 314–319.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

OpenAI. 2022. text-davinci-003: A Variant of the GPT-3 Language Model. `https://openai.com`. Accessed: February 20, 2024.

OpenAI. 2024. ChatGPT: A Large-Scale Transformer-Based Language Model. `https://openai.com/research/chatgpt`. Accessed: February 20, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Mike Schuster and Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for turing test in the age of neural text generation. *CoRR*, abs/2109.13296.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Appendix A

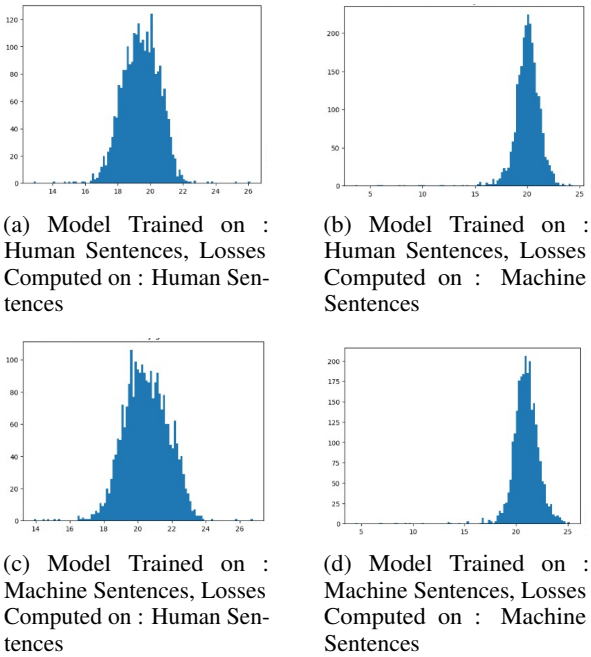
A. Setup

In this study, we implemented a methodology aimed at distinguishing human-generated sentences from machine-generated ones within a training dataset. To achieve this, we initially segregated the dataset into two distinct subsets: one containing human-generated sentences and the other comprising machine-generated ones. Subsequently, we trained separate models utilizing these segregated datasets. Specifically, we employed two distinct models for this task : i) Bidirectional Long Short-Term Memory (**BiLSTM**) model, ii) **RoBERTa** model.

Following the training phase, we proceeded to evaluate the performance of both models on a validation dataset. During this evaluation, we measured the loss incurred by each model when tasked with discerning between human-generated and machine-generated sentences. This evaluation process was crucial for assessing the efficacy and generalization capabilities of the trained models in accurately distinguishing between the two types of sentences.

B. Results

The results are in form of graphs in Figure 2



(a) Model Trained on : Human Sentences, Losses Computed on : Human Sentences

(b) Model Trained on : Human Sentences, Losses Computed on : Machine Sentences

(c) Model Trained on : Machine Sentences, Losses Computed on : Human Sentences

(d) Model Trained on : Machine Sentences, Losses Computed on : Machine Sentences

Figure 2: Overall Results on Models trained on Human and Machine Generated Sentences and Losses Calculated on Human and Machine Generated Sentences

We noted a consistent pattern across both sets of models – those trained on human-generated sentences and those trained on machine-generated sentences. Specifically, we observed that the losses incurred by human-generated sentences on the validation set exhibited a wider distribution with higher variance, while the losses associated with machine-generated sentences displayed a narrower distribution with lesser variance.

This observation leads to a compelling inference regarding the predictive nature of the model losses for each type of data. The wider distribution and higher variance in losses for human-generated sentences suggest a greater level of unpredictability associated with these sentences. In contrast, the narrower distribution and lesser variance in losses for machine-generated sentences indicate a higher level of predictiveness in the model's performance on these sentences.

This finding sheds light on the inherent characteristics of human-generated versus machine-generated sentences, particularly regarding their predictability when processed by the trained models. Such insights are crucial for understanding the intricacies of model behavior and the challenges posed by different types of data in natural language processing tasks.

HW-TSC 2024 Submission for the SemEval-2024 Task 1: Semantic Textual Relatedness (STR)

Mengyao Piao, Chang Su, Yuang Li, Xiaosong Qiao, Xiaofeng Zhao, Yinglu Li, Min Zhang, Hao Yang, Dandan Tu
Huawei Translation Services Center, China
{piaomengyao1, suchang8, liyuang3, qiaoxiaosong, zhaoxiaofeng14, liyinglu, zhangmin186, yanghao30, tudandan}@huawei.com

Abstract

The degree of semantic relatedness of two units of language has long been considered fundamental to understanding meaning. In this paper, we present the system of Huawei Translation Services Center (HW-TSC) for Task 1 of SemEval 2024, which aims to automatically measure the semantic relatedness of sentence pairs in African and Asian languages. The task dataset for this task covers about 14 different languages, These languages originate from five distinct language families and are predominantly spoken in Africa and Asia. For this shared task, we describe our proposed solutions, including ideas and the implementation steps of the task, as well as the outcomes of each experiment on the development dataset. To enhance the performance, we leverage these experimental outcomes and construct an ensemble one. Our results demonstrate that our system achieves impressive performance on test datasets in unsupervised track B and ranked first place for the Punjabi language pair [1].

1 Introduction

The semantic relatedness of two units of language is the degree to which they are close in terms of their meaning (Abdalla et al., 2021). The linguistic units can be words, phrases, sentences, etc. Though our intuition of semantic relatedness is dependent on many factors such as the context of assessment, age, and socioeconomic status (Harispe et al., 2015), it is argued that a consensus can usually be reached for many pairs (Harispe et al., 2015). In the SemEval 2024 shared task 1 (Ousidhoum et al., 2024b), there are three sub-tracks — Track A: Supervised, Track B: Unsupervised, and Track C: Cross-lingual and each track involves several language pairs. Our team — Huawei Translation Services Center (HW-TSC) — participated in the Track B: Unsupervised one which covers most African and Asian language pairs and has to be developed without the use of any labeled data for semantic relatedness. In this paper, we describe HW-TSC's system for unsupervised semantic relatedness tasks, which leverages multiple pre-trained multilingual language models to capture the semantic relatedness of different language pairs. The main features of our system are as follows:

- **N-gram Chars Method**: We employ the tokenizers of two base models for this method. The first one is XLM-RoBERTa (Conneau et al., 2019), a large unsupervised cross-lingual model that extends Facebook's RoBERTa model with more languages and data. The second one is MultilingualBERT (Devlin et al., 2018), a transformers model that is pre-trained on a large multilingual corpus using self-supervised objectives. To measure the similarity between two sentences, we use their n-gram dictionaries as features and compute a similarity score based on them.

- **BERTScore Method**: This method adopts a metric, named BERTScore (Zhang* et al., 2020). It is a metric to assess the quality of the generated text. BERTScore is mainly based on the idea of computing a score from the cosine similarity of the token-level representations obtained from the BERT model for the generated and reference texts.

- **Pretrained Large Language Model Method**: We use XGLM (Lin et al., 2021), a large-scale auto-regressive language model, as the backbone of this method. XGLM is a pre-trained language model that can handle multiple languages and domains. By leveraging the powerful large language model, we can efficiently

[1] https://docs.google.com/spreadsheets/d/1KGN26MYVlfEOqooq-bzD6EBNnpl-YT5XrY9COKESS-g/edit?usp=sharing

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1634–1638

obtain the token logits and perform calculations with them.

- **Translate to English and N-gram Chars Method**: This method needs us to process data with a translation system first, which converts the data from various languages into English. After the translation, we follow the same procedure as the N-gram Chars Method, which uses the n-gram character dictionaries of the generated and the reference texts to compute a similarity score.

- **Dataset**: We utilize the original development and test dataset from SemRel2024 (Ousidhoum et al., 2024a), a novel collection of semantic relatedness datasets annotated by native speakers for 14 languages: Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish, and Telugu.

In this paper, we analyze the characteristics of the shared task and describe our solutions, which include the ideas and implementation processes. We use Sentence-BERT (Reimers et al., 2019) as our baseline for the experiment. We conduct various experiments with the base model, large language models, etc. Our model achieves the best performance for the Punjabi language pairs in the unsupervised track B. The results are encouraging for semantic relatedness, although there is still scope for improvement.

2 Related Work

Our track is unsupervised, meaning that the systems submitted by participants do not rely on any labeled data for measuring semantic relatedness or similarity between text units longer than two words in any language. Consequently, any pre-trained language models that are further fine-tuned with text similarity data, using methods such as instruct-tuning, classification, or a similarity objective, are disqualified from our track. For our baseline score, we used Sentence-BERT (Reimers et al., 2019) (SBERT), a variant of the pre-trained BERT network that employs siamese and triplet architectures to generate sentence embeddings that are semantically meaningful and comparable by cosine similarity. SBERT has been fine-tuned on natural language inference (NLI) data, resulting in sentence embeddings that surpass other state-of-the-art methods. Hence, we selected SBERT as our baseline model and obtained our baseline score.

We introduce BERTScore (Zhang* et al., 2020) secondary, which is an automatic evaluation metric for text generation. Analogously to common metrics, BERTScore computes a similarity score for each token in the candidate sentence with each token in the reference sentence. What is more, different from other matches, it computes token similarity using contextual embeddings. BERTScore correlates better with human judgments and provides stronger model selection performance than existing metrics.

3 Method

3.1 N-gram Chars Method

The n-gram method (Kondrak and Grzegorz, 2005) is a statistical method used in natural language processing (NLP) to analyze the co-occurrence of words in a given text. It involves breaking down the text into sequences of words, where each sequence contains a fixed number of words, referred to as n-grams. The most common types of n-grams are bi-grams (2-grams), tri-grams (3-grams), and quadri-grams (4-grams), but n-grams can have any length. The primary purpose of using n-gram models is to capture the statistical dependencies between words in a language. By analyzing these dependencies, n-gram models can be used for various NLP tasks, such as language modeling, text generation, machine translation, and information retrieval. In this task, we first realized this way, for tokenizing, we tried pre-trained model XLM-Roberta (XLMR) and Multilingual-BERT (MBERT) because it is a multilingual task. At last, we calculate the similarity score with two sentences' n-gram dictionary shown as algorithm1.

3.2 BERTScore Method

BERTScore is a metric for evaluating the quality of text generation, particularly for tasks like machine translation, summarization, and text completion. BERTScore leverages the pre-trained BERT model (Bidirectional Encoder Representations from Transformers) to measure the semantic and syntactic alignment between the generated text and its reference or target text. The core idea behind BERTScore is to compute a score based on the cosine similarity of token-level representations from the BERT model for the generated text and

Algorithm 1 N-gram Chars Score Method

Require: Word sequences of the two sentences Sq_a, Sq_b; there length $Len_a \leftarrow len(Sq_a)$, $Len_b \leftarrow len(Sq_b)$; N-gram window width N

Ensure: $0 < N < min(Len_a, Len_b)$

$Dict_{\{a,b\}} \leftarrow \{\}$
for $i \leftarrow 0$ **to** $Len_{\{a,b\}} - N$ **do**
  $W_{\{a,b\}i} \leftarrow Sq_{\{a,b\}}[i : i + N]$
  if $W_{\{a,b\}i}$ **not in** $Dict_{\{a,b\}}$ **then**
    $Dict_{\{a,b\}}\}[W_{\{a,b\}i}] = 1$
  else
    $state \leftarrow Dict_{\{a,b\}}[W_{\{a,b\}i}] + 1$
    $Dict_{\{a,b\}}[W_{\{a,b\}i}] \leftarrow state$
  end if
end for
$same \leftarrow 0$
for all key **from** $Dict_a$ **do**
  if key **is in** $Dict_b$ **then**
    $count \leftarrow min(Dict_a[key], Dict_b[key])$
    $same \leftarrow same + count$
  end if
end for
$score \leftarrow \frac{2 \times same}{Len_a + Len_b - 2N + 2}$
return $score$

the reference text. Additionally, BERTScore does not require training or tuning and is based on a publicly available pre-trained model. This makes it a useful and practical tool for evaluating the quality of generated text in various natural language processing tasks. Therefore, we calculate the score with the leverage of BERTScore.

3.3 Pretrained Large Language Model Method

Different from the method above, we take advantage of the pre-trained large language model to obtain the logits of the token and compute the score. XGLM is an open-source general language model pre-training framework[2]. The model architecture is general and can be easily extended, supporting various model scales and task-specific architectures. XGLM uses a Transformer-based architecture, after pre-training XGLM learns language structure and grammatical rules, and can generate high-quality natural language text. All in all, XGLM is a flexible and powerful general language model pre-training framework, that supports only Chinese and English. Therefore, we first use the model on the English task to get the logits of the token. Then try to calculate the sum, mean, and half of the logits in proper order.

[2] https://huggingface.co/docs/transformers/main/en/model_doc/xglm

3.4 Translate to English and N-gram Chars Method

Though XLMR and MBERT can support multiple languages, if we look closely at the training data we can see that most of it is in English. Our track mainly faced 14 different African and Asian languages, in order to satisfy our track more, we took advantage of our team to process the data with a translation system to make the data from African and Asian languages into English. And then get the logits of the token as well as the last one.

4 Experiments Results

In the beginning, we applied the following three methods to the English development dataset: N-gram Chars Method with XLMR and MBERT, BERTScore Method, and Pre-trained Large Language Model Method with XGLM to calculate the sum, mean, and half of the logits. See Table 1 Different methods on English development dataset. We can see Ngram-XLMR, Ngram-MBERT, and BERTScore got really impressive performance than every method on XGLM, though XGLM is a large language model and can generate high-quality natural language text almost all the methods with XGLM are below 0.5 in the score.

Method-Model	Score
Ngram-XLMR	0.651
Ngram-MBERT	0.604
BERTScore	0.650
sum-XGLM	0.091
mean-XGLM	0.314
half-XGLM	0.211

Table 1: Different method on English development dataset

Afterwards, we use these methods on the Afrikaans development dataset. What's more, we add Translate to English and N-gram Chars Based method. See Table 2 Different methods on Afrikaans development dataset. we can conclude that Ngram-XLMR and BERTScore still perform better than other methods. What is more, the Translate to English and N-gram Chars Based method did not bring us too many surprises. The table shows that the methods that translate to English are

Method-Model	Score
Ngram-XLMR	0.475
Ngram-MBERT	-0.170
BERTScore	0.102
eng-Ngram-XLMR	-0.171
eng-Ngram-MBERT	0.014
eng-BERTScore	0.102

Table 2: Different method on Afrikaans development dataset

almost all below the methods that did not.

Ngram-XLMR ratio	Score
0	0.650
0.1	0.689
0.2	0.690
0.3	0.689
0.4	0.685
0.5	0.680
0.6	0.676
0.7	0.674
0.8	0.673
0.9	0.672
1	0.651

Table 3: Different ensemble ratio with Ngram-XLMR and BERTScore on English development dataset

Ngram-XLMR ratio	Score
0	0.175
0.1	0.126
0.2	0.106
0.3	0.093
0.4	0.088
0.5	0.084
0.6	0.082
0.7	0.080
0.8	0.080
0.9	0.080
1	0.099

Table 4: Different ensemble ratio with Ngram-XLMR and BERTScore on Punjabi development dataset

From the experiments above, we can see N-gram Chars Based with XLMR and MBERT, BERTScore Based can always get better performance in English and Afrikaans. Will they get a better performance in the other 12 languages? See Table 5 this shows three methods and a Baseline on all language development datasets. To compare with the result from the three methods and baseline, we can see Ngram-XLMR and BERTScore always get better scores in all languages.

At last, we make other experiments to ensemble the results of Ngram-XLMR and BERTScore methods to find out if this way can bring us better performance. We make Ngram-XLMR with ratio A, and BERTScore method with ratio (1-A). See Table 3 Different ensemble ratio with Ngram-XLMR and BERTScore on English development dataset. See Table 4 Different ensemble ratio with Ngram-XLMR and BERTScore on Punjabi development dataset. We can see that the ensemble way may or may not improve the performance.

5 Conclusion

This paper describes HW-TSC's unsupervised system for Semantic Textual Relatednes shared task held in SemEval 2024 Task 1 and also presents the design, the data, and the results. The participants of the shared task were provided with a collection of unsupervised datasets in multiple languages. The shared task is challenging, partly due to the unsupervised development data, and can not use models that have fine-tuned with text similarity data whether through instruct-tuning (e.g., BLOOMZ (Muennighoff et al., 2022)), classification, or a similarity objective (like SBERT). Our system uses three base models with the dataset and carries out comprehensive experiments with different pre-trained models and methods. Finally, our system achieved the 1st best performance in the Punjabi language. For some of the problems reflected in this task, there is still a lot of research space. In the future, we will investigate the transfer method to transfer the knowledge of one language to multiple languages to improve efficiency and we plan to leverage other multiple languages model's skills.

References

Abdalla, Mohamed, Vishnubhotla, Krishnapriya, Mohammad, and Saif M. 2021. What makes sentences semantically related: A textual relatedness dataset and empirical study. *arXiv preprint arXiv:2110.04845*.

Conneau, Alexis, Khandelwal, Kartikay, Goyal, Naman, Chaudhary, Vishrav, Wenzek, Guillaume, Guzmán, Francisco, Grave, Edouard, Ott, Myle, Zettlemoyer, Luke, Stoyanov, and Veselin. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Language	Ngram-XLMR	BERTScore	Ngram-MBERT	SBERT(Baseline)
eng	0.651	0.650	0.604	0.758
afr	0.475	0.102	-0.170	0.639
amh	0.630	0.085	0.630	0.650
arb	0.214	0.226	0.214	0.402
arq	0.487	0.420	0.408	0.296
ary	0.565	0.524	0.462	0.460
hau	0.325	0.240	0.325	0.382
hin	0.585	0.684	0.581	0.613
ind	0.490	0.468	0.501	0.445
kin	0.115	0.010	0.130	0.323
pan	0.099	0.175	0.099	0.173

Table 5: Top three methods and Baseline on all languages development dataset

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, and Kristina. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Harispe, Sébastien, Ranwez, Sylvie, Janaqi, Stefan, Montmain, and Jacky. 2015. *Semantic similarity from natural language and ontology analysis*. Springer.

Kondrak and Grzegorz. 2005. N-gram similarity and distance. In *International symposium on string processing and information retrieval*, pages 115–126. Springer.

Lin, Xi Victoria, Mihaylov, Todor, Artetxe, Mikel, Wang, Tianlu, Chen, Shuohui, Simig, Daniel, Ott, Myle, Goyal, Naman, Bhosale, Shruti, Du, Jingfei, et al. 2021. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*.

Muennighoff, Niklas, Wang, Thomas, Sutawika, Lintang, Roberts, Adam, Biderman, Stella, Scao, Teven Le, Bari, M Saiful, Shen, Sheng, Yong, Zheng-Xin, Schoelkopf, Hailey, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Reimers, Nils, Gurevych, and Iryna. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

KnowComp at SemEval-2024 Task 9: Conceptualization-Augmented Prompting with Large Language Models for Lateral Reasoning

Weiqi Wang, Baixuan Xu, Haochen Shi, Jiaxin Bai, Qi Hu, Yangqiu Song
Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China
{wwangbw, bxuan, hshiah, jbai, qhuaf, yqsong}@cse.ust.hk

Abstract

Lateral thinking is essential in breaking away from conventional thought patterns and finding innovative solutions to problems. Despite this, language models often struggle with reasoning tasks that require lateral thinking. In this paper, we present our system for SemEval-2024 Task 9's BrainTeaser challenge, which requires language models to answer brain teaser questions that typically involve lateral reasoning scenarios. Our framework is based on large language models and incorporates a zero-shot prompting method that integrates conceptualizations of automatically detected instances in the question. We also transform the task of question answering into a declarative format to enhance the discriminatory ability of large language models. Our zero-shot evaluation results with ChatGPT indicate that our approach outperforms baselines, including zero-shot and few-shot prompting and chain-of-thought reasoning. Additionally, our system ranks ninth on the official leaderboard, demonstrating its strong performance.

1 Introduction

Recently, the Natural Language Processing (NLP) community has witnessed remarkable advancements driven by large language models, such as GPT-3.5 (OpenAI, 2022) and GPT4 (OpenAI, 2023), that demonstrated impressive capabilities in tasks like text generation (Chung et al., 2023; Maynez et al., 2023; Maiorino et al., 2023), translation (Mu et al., 2023; Bawden and Yvon, 2023; Zhang et al., 2023), reasoning (Huang and Chang, 2023; Chan et al., 2024; Gaur and Saunshi, 2023; Ho et al., 2023; Shi et al., 2023), complex reasoning (Bai et al., 2023; Fang et al., 2024), analogical understanding (Cheng et al., 2023; Ye et al., 2024)and sentiment analysis (Carneros-Prado et al., 2023; Deng et al., 2023). However, these models predominantly rely on conventional sequential thinking, often struggling to exhibit the creativity and innovative problem-solving abilities that humans possess. This limitation has spurred researchers to explore the realm of lateral thinking within the NLP domain (Veale and Li, 2013).

Lateral thinking, a concept popularized by De Bono (1970), refers to the ability to break free from established thought patterns and approach problems from unconventional angles. It encourages the exploration of unorthodox ideas, perspectives, and solutions, leading to breakthroughs and the discovery of new opportunities that may have otherwise remained hidden (Lawrence et al., 2016). Harnessing the power of lateral thinking can significantly enhance the capabilities of language models, enabling them to tackle complex, non-linear challenges by thinking "outside the box." However, engaging in this type of reasoning presents a significant challenge, as it demands the ability to contradict common knowledge—a skill highly valued by cutting-edge language models like ChatGPT (OpenAI, 2022) and GPT4 (OpenAI, 2023). Challenging traditional modes of commonsense reasoning poses a serious obstacle for these language models, as it requires them to set aside their inherent strengths and approach the problem from a different perspective.

In light of this direction, Jiang et al. (2023) have recently introduced BrainTeaser, a human-curated benchmark that evaluates the lateral thinking ability of language models. This benchmark encompasses sentence and word puzzles in a question-answering format that challenge common sense, demanding language models to demonstrate innovative thinking in order to provide accurate and insightful responses. The findings of this study expose a significant disparity in the lateral thinking capacities of even large-scale language models, including those augmented with commonsense knowledge (Wang et al., 2023a), when compared to human performance. This gap in accuracy exceeds 40%, emphasizing the necessity for novel

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1639–1645

approaches to enhance the reasoning capabilities of language models.

We propose a new approach to enhance the lateral thinking capability of language models by applying conceptualization (He et al., 2022). Conceptualization is the process of abstracting instances into high-level concepts, which introduces abstract knowledge associated with the concept for the instance (Tenenbaum et al., 2011). Our method involves instructing ChatGPT to perform conceptualization over the premises in the question via a step-by-step process that identifies instances, conceptualizes them into concepts, generates relevant abstract knowledge, and merges them back into the prompt. To make the judgment less biased among choices, we transform the questions into declarative formats. We test our framework with ChatGPT (OpenAI, 2022) in a zero-shot manner, where no training data is used. Our experiment results show that our framework achieves an overall accuracy of 78.3% for sentence puzzles and 85.4% for word puzzles, ranking ninth and eighth in the official leaderboard, respectively.

2 Related Works

2.1 Lateral Reasoning

Lateral reasoning, also known as "thinking outside the box," has garnered significant attention in cognitive psychology and educational research (Evans and Alderson, 2000). Over the past decades, researchers have explored various aspects of lateral reasoning, aiming to understand its underlying processes and develop effective strategies to enhance individuals' lateral thinking abilities (Millar and Taylor, 1995). It is known to be challenging as such type of reasoning usually defies commonsense knowledge, which is knowledge about facts in the world that is typically shared among individuals (Mueller, 2014; Fang et al., 2021b,a). In the domain of NLP, Jiang et al. (2023) are the first to construct evaluation benchmarks that evaluate such cognitive ability. They formulate the task as a question-answering task and design a data collection protocol to crawl sentence puzzles and word puzzles from the web with quality filtering. Experiment results on various language models show the difficulty of their collected dataset.

2.2 Conceptualization

Conceptualization aims to abstract a set of entities or events into a general concept, thereby forming abstract commonsense knowledge within its original context (Murphy, 2004). Existing works primarily focused on entity-level conceptualization (Durme et al., 2009; Song et al., 2011, 2015; Liu et al., 2022), with He et al. (2022) pioneering the construction of an event conceptualization benchmark by extracting concepts for social events from WordNet (Miller, 1995) synsets and Probase (Wu et al., 2012). Wang et al. (2023b,a) further proposed a semi-supervised framework for conceptualizing CSKBs and demonstrated that abstract knowledge can enhance commonsense inference modeling and question answering. Wang et al. (2024) proposed distilling such type of knowledge from large language models to improve commonsense reasoning. Wang et al. (2023c) and Yu et al. (2023) also leveraged similar method to acquire abstract knowledge as high-level knowledge representation. In this paper, we share similar aspirations from previous works and leverage the power of conceptualization to assist large language models in performing lateral reasoning.

	Sentence Puzzle	Word Puzzle
#Data	120	96

Table 1: Number of data in the testing set of the BrainTeaser (Jiang et al., 2023) benchmark.

3 Task Definition and Dataset

We follow the identical task definition as proposed by Jiang et al. (2023), where each data entry can be viewed as a Question-Answering (QA) task. In each QA pair, the question describes a specific context or puzzle, and the answer serves as the lateral explanation or solution to the puzzle. The goal is to find an explanation that supports and does not contradict a given set of premises (P), which includes explicitly stated clauses and implicitly derived clauses through default commonsense inferences or associations. The set of premises (P) plays a crucial role in the puzzle. It encompasses the atomic premise set, which includes explicitly stated clauses (p_1, p_2, p_3) provided by the context, as well as implicit clauses (p_4, p_5) obtained through default commonsense inferences or associations. These implicit premises can sometimes lead to incorrect assumptions or constraints that hinder finding the correct solution (Bar-Hillel et al., 2018). The puzzle is presented in a multiple-choice format,

where the answer choices represent potential explanations or solutions. This format is chosen to make the task more amenable to automated evaluation and facilitate human comprehension.

We use the dataset presented by Jiang et al. (2023, 2024) as our evaluation benchmark and follow the the original released split of data. Since we approach this task by following a zero-shot manner, no training and validation data is used. As shown in Table 1, there are 120 sentence puzzles and 96 word puzzles in the testing set. On average, the questions in this dataset consist of 34.88 tokens, while the corresponding answers have an average length of 9.11 tokens.

4 Method

In this section, we introduce our proposed method. Our method can be divided into three steps: (1) automatically identify instances in the premises in the question and conceptualize them; (2) transform the QA pair into declarative statements; and (3) Prompt ChatGPT in a zero-shot manner to obtain its prediction.

4.1 Conceptualization Augmentation

Our approach to conceptualization follows the method proposed by Wang et al. (2024). First, we provide ChatGPT with a question from the BrainTeaser QA pairs and instruct it to identify relevant keywords and instances in the question. Specifically, we ask it to focus on instances that are pertinent to the question at hand. Next, we utilize the prompt from Wang et al. (2024) to guide ChatGPT in generating conceptualizations for the identified instances. We also instruct ChatGPT to generate abstract knowledge that is relevant to the context of the question. Both the generated conceptualizations and abstract knowledge are integrated into the prompts to assist in the reasoning process. For example, consider the question "A man shaves everyday, yet keeps his beard long" in a sentence puzzle. ChatGPT identifies *shave* and *beard* as the two key instances. The instance "shave" is then conceptualized to "shaving," which further implies that *shaving causes a man's beard go short.*

4.2 Declarative Transformation

We then convert each puzzle into a declarative format and modifying the task to involve selecting the most plausible statement from the options, rather than the traditional question-and-answer format. To achieve this, we present ChatGPT with the question and one of the potential answers, and instruct it to generate a declarative statement that conveys the same meaning as the given question and answer with minimal alterations. For instance, consider the question "In a small village, two farmers are working in their fields - a diligent farmer and a lazy farmer. The hardworking farmer is the son of the lazy farmer, but the lazy farmer is not the father of the hardworking farmer. Can you explain this unusual relationship?" and one of the options, "The lazy farmer is his mother." In response, ChatGPT produces the statement "In a small village, there are two farmers working in their fields - a diligent farmer and a lazy farmer. The hardworking farmer is the son of the lazy farmer, but the lazy farmer is not the father of the hardworking farmer. This peculiar relationship can be clarified by asserting that the lazy farmer is, in fact, the mother of the hardworking farmer."

4.3 Zero-shot Prompting

Finally, we prompt ChatGPT again to ask it to select the most plausible one from the given three statements. For each statement, we also append the derived conceptualizations and associated abstract knowledge into the statement such that they can also be considered during the selection process. We also ask ChatGPT to focus on whether the abstract knowledge has any conflict to the statement presented, which aims at identifying conflicts between commonsense knowledge and the presented statement.

5 Experiments

In this section, we present details of experiments we conducted on the BrainTeaser benchmark.

5.1 Setup

We access ChatGPT through Microsoft Azure APIs[1]. The code of the accessed version for ChatGPT is `gpt-35-turbo-20230515`. The maximum generation length is set to 100 tokens and the temperature is set to 1.0. All other hyperparameters remain unchanged as default. We experiment with three random seeds and report the best performances achieved according to the leaderboard's ranking. For the evaluation metric, we keep using accuracy as the metric and also evaluate the puzzles in instance-based and group-based fashions.

[1] https://azure.microsoft.com/en-us/products/ai-services/

Category	Model	Instance-based			Group-based		Overall
		Original	Semantic	Context	Ori & Sem	Ori & Sem & Con	
				Sentence Puzzle			
Random	-	25.52	24.88	22.81	5.58	1.44	24.40
Instruction	FlanT5(11B; zero-shot)	33.49	31.58	36.84	22.01	11.00	33.97
	FlanT5(11B; two-shot)	37.80	33.49	38.76	26.79	13.40	36.68
	FlanT5(11B; four-shot)	38.28	34.45	41.15	26.79	13.40	37.96
	FlanT5(11B; six-shot)	38.28	34.45	41.63	27.27	13.88	38.12
	FlanT5(11B; eight-shot)	38.76	33.01	41.63	26.79	14.35	37.80
	T0(11B)	22.01	22.01	29.67	16.27	11.00	24.56
	T0P(11B)	23.92	22.49	34.93	17.70	11.96	27.11
	T0PP(11B)	26.32	27.27	37.80	19.14	11.96	30.46
	ChatGPT(zero-shot)	60.77	59.33	67.94	50.72	39.71	62.68
	ChatGPT(two-shot)	61.72	60.77	68.90	51.67	40.67	63.80
	ChatGPT(four-shot)	59.33	55.98	62.20	47.85	32.06	59.17
	ChatGPT(six-shot)	60.29	59.81	66.51	51.20	40.19	62.20
	ChatGPT(eight-shot)	63.16	62.68	67.46	54.55	44.02	64.43
Commonsense	RoBERTa-L(CSKG)	35.41	36.84	44.98	28.71	18.18	39.07
	CAR	10.53	10.53	11.48	5.74	2.39	10.85
Ours	ChatGPT w. Concept.	**82.50**	**77.50**	**75.00**	**72.50**	**62.50**	**78.30**
Human*	-	90.74	90.74	94.44	90.74	88.89	91.98
				Word Puzzle			
Random	-	26.02	27.85	22.51	7.32	1.83	25.34
Instruction	FlanT5(11B; zero-shot)	42.68	32.93	43.90	28.66	20.12	39.84
	FlanT5(11B; two-shot)	44.51	34.76	45.73	30.49	18.90	41.67
	FlanT5(11B; four-shot)	43.29	35.98	47.56	30.49	20.73	42.28
	FlanT5(11B; six-shot)	44.51	36.59	47.56	29.88	17.68	42.89
	FlanT5(11B; eight-shot)	45.73	33.54	46.95	27.44	16.46	42.07
	T0(11B)	17.07	14.02	23.17	9.76	6.10	18.09
	T0P(11B)	28.66	26.22	34.15	19.51	12.80	29.67
	T0PP(11B)	33.54	31.10	39.63	20.12	10.98	34.76
	ChatGPT(zero-shot)	56.10	52.44	51.83	43.90	29.27	53.46
	ChatGPT(two-shot)	55.49	53.66	51.22	44.51	30.49	53.46
	ChatGPT(four-shot)	54.27	53.66	51.83	43.90	28.05	53.25
	ChatGPT(six-shot)	56.71	51.83	**54.27**	45.12	28.66	54.27
	ChatGPT(eight-shot)	58.54	56.71	54.27	48.17	34.76	56.50
Commonsense	RoBERTa-L(CSKG)	18.90	16.46	30.49	12.80	6.10	21.95
	CAR	38.41	31.10	20.12	26.22	6.10	29.88
Ours	ChatGPT w. Concept.	**84.40**	**90.60**	**81.20**	**84.40**	**65.60**	**85.40**
Human*	-	91.67	91.67	91.67	91.67	89.58	91.67

Table 2: Main zero-shot results over two BrainTeaser subtasks across all models in all metrics: Ori = Original, Sem = Semantic, Con = Context, Concept = Conceptualization. The best performance among all models is in bold, and the second-best performance is underlined. Most of the results are reported by Jiang et al. (2023).

5.2 Baselines

For baselines, we largely follow Jiang et al. (2023) and use the officially reported results as baselines. These include instruction-based language models such as ChatGPT (OpenAI, 2022), T0 (Sanh et al., 2022), and FlanT5 (Chung et al., 2022), which were evaluated in a zero setting using specific instruction templates. In addition, commonsense models were also evaluated, including RoBERTa-L (CSKG; Ma et al., 2021) and CAR (Wang et al., 2023a), which were enhanced with commonsense knowledge and achieved impressive zero-shot performance on multiple tasks. The models were evaluated using a scoring method defined in previous studies and the choice with the highest score is selected. Meanwhile, we also report the performances of ChatGPT in a few-shot setting with up to eight shots.

5.3 Results and Analysis

Table 2 presents the results of our study. Our method significantly improves the performance of ChatGPT, outperforming all baselines. In fact, it surpasses all large language models in a zero-shot scenario and even outperforms ChatGPT itself with eight-shot prompting. For sentence puzzles, we observe an overall improvement of 13.87%, while for word puzzles, there is a 28.90% improvement. However, our method still falls short of human performance, indicating room for further improvement. Interestingly, we notice a larger improvement in word puzzles compared to sentence puzzles. This gain may be attributed more to our declarative trans-

formation than to conceptualization, which theoretically offers little help in solving word puzzles.

6 Conclusion

In conclusion, this paper describes the solution by the KnowComp group to task 9 of SemEval-2024. Our method tackles the task of lateral thinking by leveraging the framework of conceptualization, which is a traditional reasoning method performed by humans, to assist large language models in answering brain teaser questions in a zero-shot manner. Experiment results show the superiority of our method, outperforming all previous zero-shot baselines with the same large language model as the backbone.

Acknowledgements

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20), and the GRF (16211520 and 16205322) from RGC of Hong Kong. We also thank the support from the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).

References

Jiaxin Bai, Xin Liu, Weiqi Wang, Chen Luo, and Yangqiu Song. 2023. Complex query answering on eventuality knowledge graph with implicit logical constraints. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Maya Bar-Hillel, Tom Noah, and Shane Frederick. 2018. Learning psychology from riddles: The case of stumpers. *Judgment and Decision Making*, 13(1):112–122.

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*, pages 157–170. European Association for Machine Translation.

David Carneros-Prado, Laura Villa, Esperanza Johnson, Cosmin C. Dobrescu, Alfonso Barragán, and Beatriz García-Martínez. 2023. Comparative study of large language models as emotion and sentiment analysis systems: A case-specific analysis of GPT vs. IBM watson. In *Proceedings of the 15th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2023) - Volume 2, Riviera Maya, Mexico, 28-29 November, 2023*, volume 842 of *Lecture Notes in Networks and Systems*, pages 229–239. Springer.

Chunkit Chan, Jiayang Cheng, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2024. Exploring the potential of chatgpt on sentence level relations: A focus on temporal, causal, and discourse relations. In *Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024*, pages 684–721. Association for Computational Linguistics.

Jiayang Cheng, Lin Qiu, Tsz Ho Chan, Tianqing Fang, Weiqi Wang, Chunkit Chan, Dongyu Ru, Qipeng Guo, Hongming Zhang, Yangqiu Song, Yue Zhang, and Zheng Zhang. 2023. Storyanalogy: Deriving story-level analogies from large language models to unlock analogical understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 11518–11537. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

John Joon Young Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 575–593. Association for Computational Linguistics.

Edward De Bono. 1970. Lateral thinking. *New York*, page 70.

Xiang Deng, Vasilisa Bashlovkina, Feng Han, Simon Baumgartner, and Michael Bendersky. 2023. Llms to the moon? reddit market sentiment analysis with large language models. In *Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 1014–1019. ACM.

Benjamin Van Durme, Phillip Michalak, and Lenhart K. Schubert. 2009. Deriving generalized knowledge from corpora using wordnet abstraction. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 808–816. The Association for Computer Linguistics.

Kenneth E Evans and Andrew Alderson. 2000. Auxetic materials: functional materials and structures from lateral thinking! *Advanced materials*, 12(9):617–628.

Tianqing Fang, Zeming Chen, Yangqiu Song, and Antoine Bosselut. 2024. Complex reasoning over logical queries on commonsense knowledge graphs. *arXiv preprint arXiv:2403.07398*.

Tianqing Fang, Weiqi Wang, Sehyun Choi, Shibo Hao, Hongming Zhang, Yangqiu Song, and Bin He. 2021a. Benchmarking commonsense knowledge base population with an effective evaluation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8949–8964. Association for Computational Linguistics.

Tianqing Fang, Hongming Zhang, Weiqi Wang, Yangqiu Song, and Bin He. 2021b. DISCOS: bridging the gap between discourse knowledge and commonsense knowledge. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2648–2659. ACM / IW3C2.

Vedant Gaur and Nikunj Saunshi. 2023. Reasoning in large language models through symbolic math word problems. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5889–5903. Association for Computational Linguistics.

Mutian He, Tianqing Fang, Weiqi Wang, and Yangqiu Song. 2022. Acquiring and modelling abstract commonsense knowledge via conceptualization. *CoRR*, abs/2206.01532.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14852–14882. Association for Computational Linguistics.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1049–1065. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14317–14332. Association for Computational Linguistics.

Moyra Lawrence, Sylvain Daujat, and Robert Schneider. 2016. Lateral thinking: how histone modifications regulate gene expression. *Trends in Genetics*, 32(1):42–56.

Jingping Liu, Tao Chen, Chao Wang, Jiaqing Liang, Lihan Chen, Yanghua Xiao, Yunwen Chen, and Ke Jin. 2022. Vocsk: Verb-oriented commonsense knowledge mining with taxonomy-guided induction. *Artif. Intell.*, 310:103744.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13507–13515. AAAI Press.

Antonio Maiorino, Zoe Padgett, Chun Wang, Misha Yakubovskiy, and Peng Jiang. 2023. Application and evaluation of large language models for the generation of survey questions. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 5244–5245. ACM.

Joshua Maynez, Priyanka Agrawal, and Sebastian Gehrmann. 2023. Benchmarking large language model capabilities for conditional generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9194–9213. Association for Computational Linguistics.

BJ Millar and NG Taylor. 1995. Lateral thinking: the management of missing upper lateral incisors. *British Dental Journal*, 179(3):99–106.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Yongyu Mu, Abudurexiti Reheman, Zhiquan Cao, Yuchun Fan, Bei Li, Yinqiao Li, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2023. Augmenting large language model translators via translation memories. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10287–10299. Association for Computational Linguistics.

Erik T Mueller. 2014. *Commonsense reasoning: an event calculus based approach*. Morgan Kaufmann.

Gregory Murphy. 2004. *The big book of concepts*. MIT press.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Haochen Shi, Weiqi Wang, Tianqing Fang, Baixuan Xu, Wenxuan Ding, Xin Liu, and Yangqiu Song. 2023. QADYNAMICS: training dynamics-driven synthetic QA diagnostic for zero-shot commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15329–15341. Association for Computational Linguistics.

Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. 2011. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2330–2336. IJCAI/AAAI.

Yangqiu Song, Shusen Wang, and Haixun Wang. 2015. Open domain short text conceptualization: A generative + descriptive modeling approach. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 3820–3826. AAAI Press.

Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285.

Tony Veale and Guofu Li. 2013. Creating similarity: Lateral thinking for vertical similarity judgments. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 660–670. The Association for Computer Linguistics.

Weiqi Wang, Tianqing Fang, Wenxuan Ding, Baixuan Xu, Xin Liu, Yangqiu Song, and Antoine Bosselut. 2023a. CAR: conceptualization-augmented reasoner for zero-shot commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 13520–13545. Association for Computational Linguistics.

Weiqi Wang, Tianqing Fang, Chunyang Li, Haochen Shi, Wenxuan Ding, Baixuan Xu, Zhaowei Wang, Jiaxin Bai, Xin Liu, Jiayang Cheng, Chunkit Chan, and Yangqiu Song. 2024. CANDLE: iterative conceptualization and instantiation distillation from large language models for commonsense reasoning. *CoRR*, abs/2401.07286.

Weiqi Wang, Tianqing Fang, Baixuan Xu, Chun Yi Louis Bo, Yangqiu Song, and Lei Chen. 2023b. CAT: A contextualized conceptualization and instantiation framework for commonsense reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13111–13140. Association for Computational Linguistics.

Zhaowei Wang, Haochen Shi, Weiqi Wang, Tianqing Fang, Hongming Zhang, Sehyun Choi, Xin Liu, and Yangqiu Song. 2023c. Abspyramid: Benchmarking the abstraction ability of language models with a unified entailment graph. *CoRR*, abs/2311.09174.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 481–492. ACM.

Xiao Ye, Andrew Wang, Jacob Choi, Yining Lu, Shreya Sharma, Lingfeng Shen, Vijay Tiyyala, Nicholas Andrews, and Daniel Khashabi. 2024. Analobench: Benchmarking the identification of abstract and long-context analogies. *CoRR*, abs/2402.12370.

Changlong Yu, Weiqi Wang, Xin Liu, Jiaxin Bai, Yangqiu Song, Zheng Li, Yifan Gao, Tianyu Cao, and Bing Yin. 2023. Folkscope: Intention knowledge graph construction for e-commerce commonsense discovery. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1173–1191. Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.

HW-TSC at SemEval-2024 Task 9: Exploring Prompt Engineering Strategies for Brain Teaser Puzzles Through LLMs

Yinglu Li, Yanqing Zhao, Min Zhang, Yadong Deng, Aiju Geng, Xiaoqin Liu, Mengxin Ren, Yuang Li, Chang Su, Xiaofeng Zhao, Xiaosong Qiao, Ming Zhu, Yilun Liu, Mengyao Piao,Feiyu Yao, Shimin Tao, Hao Yang, Yanfei Jiang
Huawei Translation Services Center, Beijing, China
{liyinglu, zhaoyanqing, zhangmin186, yanghao30}@huawei.com

Abstract

Large Language Models (LLMs) have demonstrated impressive performance on many Natural Language Processing (NLP) tasks. However, their ability to solve more creative, lateral thinking puzzles remains relatively unexplored. In this work, we develop methods to enhance the lateral thinking and puzzle-solving capabilities of LLMs. We curate a dataset of word-type and sentence-type brain teasers requiring creative problem-solving abilities beyond commonsense reasoning. We first evaluate the zero-shot performance of models like GPT-3.5 and GPT-4 on this dataset. To improve their puzzle-solving skills, we employ prompting techniques like providing reasoning clues and chaining multiple examples to demonstrate the desired thinking process. We also fine-tune the state-of-the-art Mixtral 7x8b LLM on our dataset. Our methods enable the models to achieve strong results, securing 2nd and 3rd places in the brain teaser task. Our work highlights the potential of LLMs in acquiring complex reasoning abilities with the appropriate training. The efficacy of our approaches opens up new research avenues into advancing lateral thinking and creative problem-solving with AI systems.

1 Introduction

In recent years, the advent of advanced language models has revolutionized the field of NLP, steering research towards challenges that necessitate intricate and implicit reasoning processes akin to human commonsense reasoning. Such tasks often require vertical thinking, an analytical and methodical approach to problem-solving. This paradigm has enjoyed substantial popularity and success within the NLP community. However, lateral thinking puzzles, which demand creative reasoning and the ability to perceive indirect or non-obvious solutions, have not been equally explored. Lateral thinking involves breaking away from conventional patterns to reveal novel insights, a feat that models based on rigid commonsense associations often struggle with.

Task	Type	Train size	Eval size	Test size
Subtask 1	Word Puzzle	396	120	96
Subtask 2	Sentence Puzzle	507	120	120

Table 1: Task dataset description

Recognizing this disparity, we introduce LLMs in "BRAINTEASER," a meticulously curated multiple-choice Question Answering (QA) task in order to evaluate LLMs' capabilities for lateral thinking. The dataset(Jiang et al., 2023b, 2024b) contains two subtasks: word and sentence brain teasers. Word puzzles are word-type brain teasers where the answer deviates from the typical meaning of the word and instead focuses on the letter composition. Sentence puzzles are sentence-type brain teasers centered around nonsensical or illogical snippets of text. The key characteristics of the dataset are described in Table 1.

In our approach, we employ the formidable GPT-4 language model to address BRAINTEASER's questions under both zero-shot and few-shot conditions, thereby assessing its inherent reasoning capabilities without and with limited context. Additionally, we leverage prompt engineering strategies and incorporate a Chain of Thought (CoT) prompting technique to enhance GPT-4's comprehension of the task requirements. This innovative methodology not only facilitates clearer demonstration of the problem-solving process but also aligns the model's reasoning with human-like thought patterns.

Examples of the word and sentence puzzle samples are provided in Tables 2 and 3, respectively. Semantic Reconstruction (SR) rephrases the original question without altering the correct answer or distractor. Context reconstruction (CR) maintains

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1646–1651

ID	Question	Choice List
WP-0	*How do you spell COW in thirteen letters?*	SEE OH DEREFORD **SEE O DOUBLE YOU.** COWCOWCOWCOWW. None of above.
WP-0_SR	*In thirteen letters, how do you spell COW?*	SEE OH DEREFORD **SEE O DOUBLE YOU.** COWCOWCOWCOWW. None of above.
WP-0_CR	*How do you spell COB in seven letters?*	COBCOBB COBBLER **SEE O BEE.** None of above.

Table 2: Dataset samples for subtask 1: word puzzles. Each choice list has four choices. The ground truth is bold.

ID	Question	Choice List
SP-48	*Why is it so cold on Christmas?*	**Because it's in December.** Because people are waiting for the New Year. Because people are celebrating. None of above.
SP-48_SR	*Why is Christmas Day so chilly?*	**Because it's in December.** Because people are waiting for the New Year. Because people are celebrating. None of above.
SP-48_CR	*Why is Independence Day so hot?*	Because people are enjoying the firework. Because people are celebrating. **Because it's in July.** None of above.

Table 3: Dataset samples for subtask 2: sentence puzzles. Each choice list has four choices. The ground truth is bold.

the reasoning path but changes both the question and answer to reflect a new situational context.

The results of our experiments are both promising and insightful. Our model achieved commendable rankings, securing 2nd and 3rd places in the task, which underscores the potential of LLMs in mastering complex, creative problem-solving tasks that extend beyond the scope of traditional commonsense reasoning. These outcomes not only validate the efficacy of our methods but also pave the way for further explorations into the untapped potential of lateral thinking in AI-driven language understanding.

2 Related work

2.1 LLM

Language is a uniquely human ability that allows us to communicate, express ourselves, and record information. In AI research, language models refer to models that can predict the next word or token in a sequence given the previous words or context. Early language models are based on statistical techniques that calculate the probability of each possible next word. These statistical language models are later superseded by neural network-based models, which can more accurately estimate the probability of the next token using deep-learning methods. The development of neural language models marks a major advance in NLP capabilities. By utilizing neural networks to model the complexities of language, today's state-of-the-art language models can generate surprisingly human-like text and show impressive language understanding abilities.

Subsequently, pretrained language models (PLM) like BERT(Devlin et al., 2018), BART(Lewis et al., 2019), and GPT2(Radford et al., 2019) are proposed. These models represent milestones in the development of language models, as they are based on the classical transformer architecture(Vaswani et al., 2023) and significantly increase the text generation capabilities of models. Initially, most of these models have relatively small sizes.

Research has shown that even by solely increasing model size while keeping model architecture similar, abilities on difficult tasks can substantially improve(Brown et al., 2020). This phenomenon of emerging abilities with scale is referred to as

emergent behavior(Wei et al., 2022). This has led to the development of LLMs which have profoundly impacted research and society. For example, the release of LLM has created much interest due to its strong text generation abilities like abstract writing and logical reasoning. This has catalyzed further research into LLMs, with models like LLaMA(Touvron et al., 2023a), LLaMA 2(Touvron et al., 2023b), Mistral 7B(Jiang et al., 2023a), GPT 4(OpenAI et al., 2023), and Mixtral 8x7B(Jiang et al., 2024a) demonstrating impressive performance on various tasks.

2.2 Prompt Engineering

Template-based prompts are among the early attempts at single-stage prompting (Paranjape et al., 2021).

However, the Chain of Thought (CoT) technique leads to more significant improvements in model capabilities (Wei et al., 2023) and attracts substantial interest. By providing a few reasoning demonstrations or "exemplars" in the prompt, CoT yields impressive performance gains. CoT also reveals LLMs' innate zero-shot reasoning abilities — simply prompting the model with "Let's think step-by-step!" enables complex inferential reasoning.

Additionally, prompt quality factors like reasoning complexity in exemplars, number of reasoning steps, and diversity of exemplars impact performance of LLM.

Since single-stage prompting may enable end-to-end reasoning, (Press et al., 2023) also explores constructing multi-stage prompts with follow-up questions and answers to provide detailed reasoning. (Jung et al., 2022) propose prompts based on trees of explanations generated abductively and recursively, e.g. X is true, because Y; Y is true, because...

(Zhou et al., 2023) find that decomposing complex questions into a series of simpler subquestions was beneficial for constructing effective prompts.

3 Method

3.1 GPT-4: From Zero-Shot to Few-Shot

Since GPT-3.5 and GPT-4 demonstrate strong performance on tasks like QA and text generation, we utilize these models to directly answer the training questions by providing the question and choice list.

For the zero-shot stage, we first explain what a word or sentence puzzle is in the prompt, presenting the question and options simultaneously. Then we use GPT-3.5 to predict answers one by one.

During this stage, we observe precision of only 17% for word puzzles on the training set. Errors frequently occur because many questions defy common knowledge, leading models to be overconfident in the "None of the above" choice. Therefore, we modify the prompt by appending "please don't choose 'None of the above', because in most cases, it is not the correct answer", increasing the precision to 66%.

We also notice some questions are too difficult for the model, such as "How many days are there in a month?". We think that providing the model with reasoning clues or demonstration may be beneficial. For this challenging sample, we guide the model to not only simply count days, but also approach the question from a new perspective — identifying which words on a calendar contain "day", like Monday and Tuesday, rather than numerals like January 1st.

A similar puzzle is "How many seconds are there in one year?". GPT-4 cannot find a correct answer if it counts the actual number of seconds in a year. We should tell it this is not to count the actual number of seconds and it should try to answer the question in another way, that is, to count the number of dates that contain second (2nd) in a year. For the hard sample "What is in front of a woman and at the end of a cow?", as an explanation, we tell GPT-4 this is a word game, and it should interpret the questions in two parts and find which letter is at the start/beginning of one word and at the end of the other word. For the question "What is at the end of a cow and in front of a woman?", we remind the model that the word woman starts with the letter "w", and the word cow ends with the letter "w". The correct answer is the letter "W". "What is at the beginning of eternity and the end of time?" For this question, the word "eternity" starts with the letter "e", and the word "time" ends with the letter "e". The correct answer is the letter "E". In this way, GPT-4 can think in the way we expect and correctly answer similar categories of brain teaser puzzles.

To address incorrectly answered examples, we identify and categorize over 20 challenging training instances to include in an extended prompt, as shown in 2. This prompt is designed to guide the model towards lateral thinking. Each illustrative example comprises the original question, choice

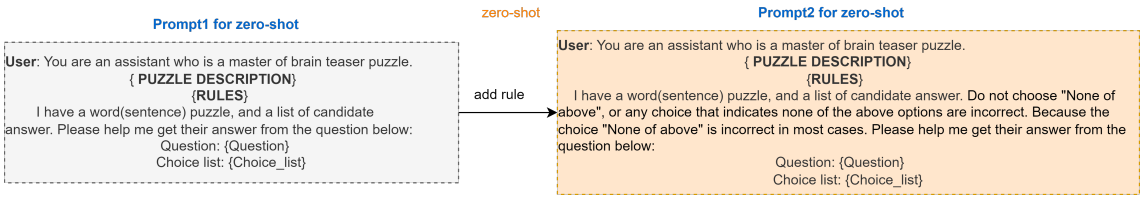


Figure 1: The left figure illustrates Prompt 1, which only provides the definition of a word or sentence puzzle before concatenating the question and choice list from the dataset. As the model tended to select 'None of the above', Prompt 2 adds a rule to avoid this answer. Both prompts are used in a zero-shot setting without examples.

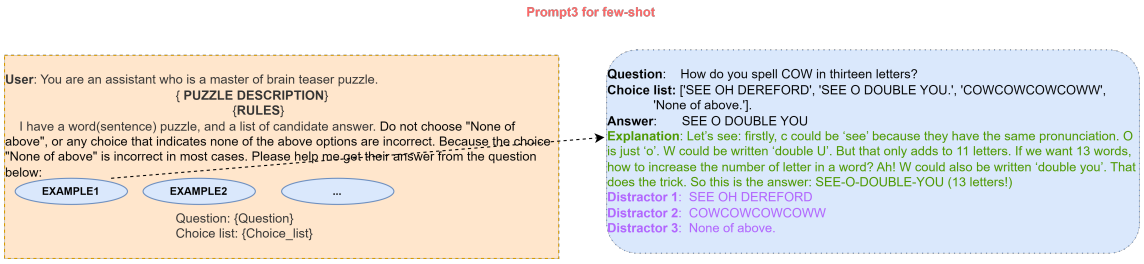


Figure 2: For the third strategy, we concatenate explanation of each example as well as distractors in the choice list. This is a few-shot strategy.

list, correct answer, and an explanatory reasoning clue extracted from the training data. Additionally, we find that supplementing each example with the three distractor options further improves GPT performance. Therefore, the full set of multiple-choice options is appended to each illustrated case. As depicted, these elements are combined to demonstrate the desired thought process.

3.2 Mixstral Fine-tuning

We also experiment with fine-tuning the Mixtral 7x8b model to predict solutions for these brain teaser puzzles. Mixtral 7x8b is a leading open-source LLM, comprised of a Mixture-of-Experts (MOE) architecture with approximately 45 billion parameters. It is regarded as state-of-the-art, outperforming models such as LLaMA 270B and GPT-3.5 on many benchmarks. Mixtral 7x8b offers both a base model and an instruct model, with the latter fine-tuned for enhanced performance on conversational tasks. Therefore, we select Mixtral-7x8b-instruct-v0.1 for fine-tuning on our dataset of around 1000 puzzle examples.

4 Experiment and Result

4.1 Experiment

Experiments are conducted on a test set to evaluate the three prompt designs introduced previously. Initially, GPT-3.5 was used to test Prompts 1 and 2 for subtask 1 (word puzzles). As the evaluation deadline approached, we switched to GPT-4 for greater efficiency. We evaluated Prompt 3 five times, and an ensemble voting strategy was adopted. Besides, we proceeded with only Prompt 3 (GPT4, with ensemble) for subtask 2's test set, omitting Prompts 1 and 2.

4.2 Result and Analysis

Experiment results on the training set are shown in Table 5, and our final results are shown in Table 6. As shown in Table 5, there is a substantial performance increase from Prompt 2 (GPT-3.5, zero-shot) to Prompt 3 (GPT-4, few-shot, with ensemble). This demonstrates the efficacy of our strategy utilizing Prompt 3 with GPT-4 in a few-shot learning setting. Besides, for the same question, GPT-4 would sometimes generate inconsistent answers or refuse to answer. To mitigate this, we ensemble the answers from 5 evaluations of each prompt by a voting strategy. This ensemble approach improves performance compared to single evaluations. Ultimately, we achieve an accuracy of 0.980 on the training subset.

	ft_mixtral_instruct
WP training set	0.21
SP training set	0.26

Table 4: Result of ft_mixtral_instruct

WP Training set (random 100 data samples)	Prompt 1 zero-shot	Prompt 2 zero-shot	Prompt 3 few-shot, with Ensemble
GPT-3.5	0.170	0.660	-
GPT-4	-	-	0.980

Table 5: Result on subtask 1: word puzzle. We use three kinds of prompt strategies on the training dataset for this subtask. We try GPT-3.5 to verify Prompt 1 and Prompt 2, and then use GPT-4 for Prompt 3. The latter strategy shows a much better performance.

SP Test Set	S_ori	S_sem	S_con	S_ori_sem	S_ori_sem_con	S_overall
	1.000	0.975	0.925	0.975	0.900	0.967
WP Test Set	W_ori	W_sem	W_con	W_ori_sem	W_ori_sem_con	W_overall
	0.969	0.938	1.000	0.938	0.938	0.969

Table 6: Final result on subtask 1 and subtask 2. We use three kinds of prompt strategies on the training dataset for the subtask. We try GPT-3.5 to verify Prompt 1 and Prompt 2, and then use GPT-4 for Prompt 3. The latter strategy shows a much better performance.

5 Conclusion

In conclusion, we demonstrate our prompt design method to enhance creative problem-solving in LLMs, enabling strong performance on brain teaser puzzles. Through prompting strategies and model fine-tuning, our methods attain 2nd and 3rd place rankings on this lateral thinking task. These results validate our techniques and highlight the potential for developing multifaceted reasoning skills in AI. Our work provides promising pathways toward more human-like language understanding and flexible thinking in natural language models. In summary, we take steps toward training AI systems capable of creative problem-solving.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024b. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023b. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, and ... 2023. Gpt-4 technical report.

Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Prompting contrastive explanations for commonsense reasoning tasks.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models.

SU-FMI at SemEval-2024 Task 5: From BERT Fine-Tuning to LLM Prompt Engineering - Approaches in Legal Argument Reasoning

Kristiyan Krumov
FMI, Sofia University
kristiyan.boyanov@gmail.com

Svetla Boytcheva
Ontotext
svetla@uni-sofia.bg
svetla.boytcheva@ontotext.com

Ivan Koytchev
FMI, Sofia University
koychev@fmi.uni-sofia.bg

Abstract

This paper presents our approach and findings for SemEval-2024 Task 5, focusing on legal argument reasoning. We explored the effectiveness of fine-tuning pre-trained BERT models and the innovative application of large language models (LLMs) through prompt engineering in the context of legal texts. Our methodology involved a combination of techniques to address the challenges posed by legal language processing, including handling long texts and optimizing natural language understanding (NLU) capabilities for the legal domain. Our contributions were validated by achieving a third-place ranking on the SemEval 2024 Task 5 Leaderboard. The results underscore the potential of LLMs and prompt engineering in enhancing legal reasoning tasks, offering insights into the evolving landscape of NLU technologies within the legal field.

1 Introduction

Legal texts, including laws, interpretations, arguments, and agreements, are commonly conveyed through writing, resulting in great amount of legal documents. Analyzing these documents, a core aspect of legal work, becomes more intricate as these collections expand. Natural language understanding (NLU) technologies offer potential assistance to legal professionals in this regard. However, their effectiveness hinges on the ability of current state-of-the-art models to adapt to diverse tasks within the legal field.

The legal argument reasoning task (Bongard et al., 2022) of SemEval-2024 represents a significant challenge in the domain of natural language processing (NLP) and an informal addition to the currently existing model evaluation benchmarks such as LexGLUE (Chalkidis et al., 2022b).

Our approach involves fine-tuning pre-trained BERT models and exploring the innovative use of large language models (LLMs) through prompt engineering to address this task.

As a result of our work, we are ranked 3-rd in the SemEval 2024 Task 5[1] Leaderboard out of 20 participating teams. The implementations of the different approaches is available on Github[2] and the fine-tuned models could be accessed in Huggingface[3].

2 Background

Task 5 of SemEval 2024 is novel NLP problem focused on legal argument reasoning within the context of U.S. civil procedure. It contributes a dataset comprised of instances each containing a general introduction to a case, a specific legal question, a proposed solution argument, and a detailed analysis explaining the applicability of the argument. This dataset aims to benchmark the performance of legal language models, posing a significant challenge due to the complexity and nuanced understanding required for legal reasoning. Instances are organized to support a binary classification task: determining the correctness of a given answer to a legal question, aimed at facilitating research on legal argument reasoning.

In the domain of text classification, conventional methodologies often employ "short encoders" such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019), which have demonstrated commendable efficacy in diverse contexts, ranging from news topic classification to sentiment analysis in movie reviews. Nevertheless, these encoders are constrained by their 512-token processing limit, rendering them less effective for analyzing extensive documents like court judgments. To circumvent this limitation, more advanced approaches, including the Hierarchical Attention Network (HAN) (Yang et al., 2016), a synergy of BERT and CNN, and the combination

[1] https://trusthlt.github.io/semeval24/
[2] https://github.com/frisibeli/semeval-2024-task5
[3] https://huggingface.co/frisibeli

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1652–1658

of XLNet with BiGRU (Chenxi et al., 2022), have been developed, enhancing the semantic understanding of longer texts. Despite these technological strides, the pursuit of an optimal algorithm that can adeptly navigate the complexities of extended documents persists.

The application of automated systems in the legal domain encounters distinct challenges, arising from the specialized language employed and the necessity for intricate multi-step reasoning over extensive texts. Furthermore, the potential of leveraging recent advancements in prompting techniques for legal domain-specific tasks remains largely unexplored. Typically, effective prompting in general NLP tasks has been noted with concise inputs, often limited to a single sentence or a small collection of sentences, accompanied by a restricted array of target labels. This underscores the ongoing quest to adapt and refine NLP techniques to meet the unique demands and intricacies of legal reasoning.

3 System Overview

After analyzing the dataset, we identified that the final system should be capable of handling relatively lengthy contexts and to perform well on reasoning and fact-checking tasks. In this section we separately introduce the different approaches we have experimented with on solving the Legal Argument Reasoning task by dividing them into methods for handling long texts and such for optimizing the NLU capabilities for the legal domain.

3.1 Handling Long Texts

Observing the distributions (Fig. 2) of the token lengths for the dataset entries we could say that a system capable of processing contexts of 2000 tokens would be sufficient to cover the majority of the cases.

3.1.1 Sliding Window (SW)

We leveraged the sliding window techniques as described in (Bongard et al., 2022), as a baseline to overcome the maximum token limit problem. We experimented with **Sliding Window Simple** and **Sliding Window Complex**

3.1.2 Transformer-based models for long text

Transformer-based models encounter difficulty processing lengthy sequences due to their self-attention operation, which exhibits quadratic scaling with sequence length. In response to this constraint, we experimented with the Longformer (Beltagy et al., 2020) model, featuring an attention mechanism that scales linearly with sequence length and increases the maximum input length to 4096 sub-word tokens, which may also improve the performance in understanding legal documents. Additionally, we experimented with Legal-RoBERTa and Legal-Longformer - pre-trained models on legal corpus introduced in (Chalkidis et al., 2023).

3.1.3 Summarizing

A different approach we tried for preprocessing lengthy texts was utilizing summarization models. By condensing extensive content into concise summaries, we not only mitigate the challenges posed by the length limitations of Transformer-based architectures but also streamline subsequent processing stages by reducing the presence of extraneous or tangential content, such as author's thoughts and remarks (Fig. 3).

As part of our solution, we examined several summarization models - BART (Lewis et al., 2019), LexRank (Erkan and Radev, 2004) and ChatGPT4[4].

3.2 Optimising NLU Capabilities for the Legal Domain

Research has demonstrated the efficacy of language model pre-training in enhancing numerous natural language understanding tasks like natural language inference (Devlin et al., 2019). In addition to learning linguistic knowledge, these models are retaining relational knowledge (Petroni et al., 2019) present in the training data which could be beneficial in solving downstream tasks in domains such as the legal one and more precisely - US Civil Procedure where the legal system is based on precedents. In this section we are going to reflect on the methods used by us to improve the performance of the system by enhancing its reasoning capabilities.

3.2.1 Pre-trained Transformer Models on Legal Corpus

As a starting point in addressing the problem, we decided to use Legal-BERT (Chalkidis et al., 2020), being the most successful baseline experiment described in the work of the organizers of the task (Bongard et al., 2022). We fine-tuned it on the task and additionally - on a custom legal dataset 3.2.2. Our contribution continued with the exploration

[4] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo

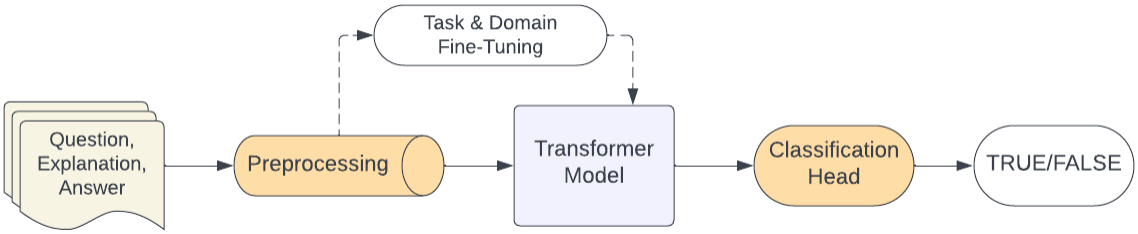


Figure 1: Transformer-based classifier system architecture

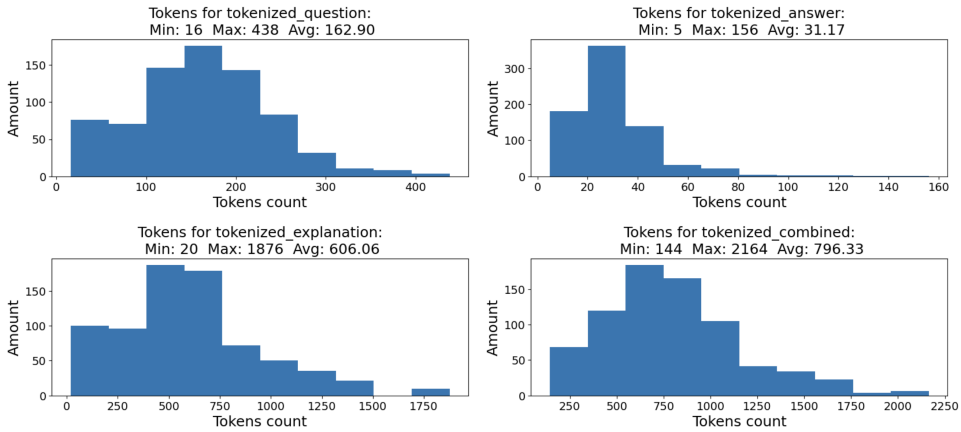


Figure 2: Token count distribution of the dataset per entry parts - Question, Explanation, Answer and Concatenated

of alternative legal transformer models: CaseHold-BERT (Zheng et al., 2021), variants of Legal-BERT (small, large), Legal-RoBERTa (Chalkidis et al., 2023) and InLegalBERT (Paul et al., 2023).

3.2.2 Fine-tuned BERT on Custom Dataset

We additionally fine-tuned the best performing models from 3.2.1 on a custom-tailored dataset of an American civil procedure data (4.3), similar to the entries from the task. The goal with this approach was to strengthen the model's relational knowledge and contextual representations of the language used in the legal domain (Petroni et al., 2019).

3.3 LLM + Legal Prompt Engineering

So far we observed the task as a supervised classification problem, where the models are trained with labeled data to classify inputs into a binary output. Another approach is to use the relatively new method of prompt engineering in combination with some of the currently best-performing generative models (Fig. 4). With prompting, there's generally no need for additional training as the model receives a prompt, which could be a question, examples of input-output pairs (few-shot learning), or task descriptions. This approach allows the model to leverage its pre-trained knowledge to produce outputs for specific tasks in a zero-shot manner, meaning it can generate correct responses without having seen examples of the specific task during its training phase. For this setup, we experimented with several types of LLMs: Mistral-7b-Instruct (Jiang et al., 2023), Llama2-70b (et al., 2023), GPT-3.5-Turbo and GPT-4 [5]; as for most of those models we performed prompt fine-tuning and Legal prompt engineering (Trautmann et al., 2022).

4 Experimental Setup

4.1 Data

For the transformer-based classifier systems (Fig. 1) we performed experiments on the SemEval 2024 Task (Bongard et al., 2022) dataset. We stratified the train partition (750 entries) into train* (88%) and train-dev (12%), ensuring that the distribution of label values was maintained. The dev partition (84 entries) and the test partition were solely utilized for validation to prevent overfitting and bias in the model.

[5] https://platform.openai.com/docs/models

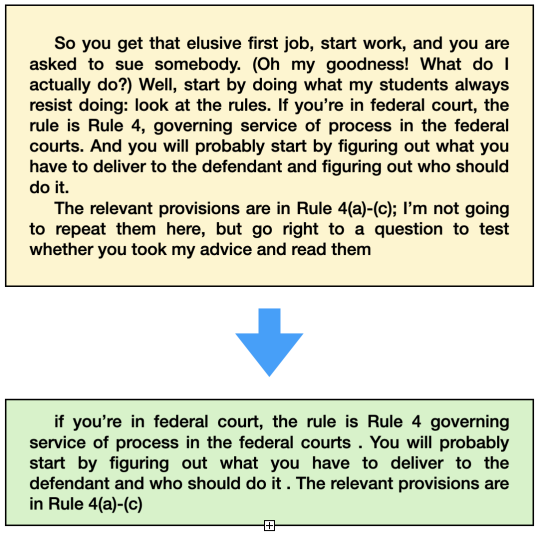


Figure 3: Preprocessing an Explanation from the dataset using T5 for summarization

On other hand, for the generative-based classifier systems (Fig. 4) we used only the dev and test partitions leveraging the generalization capabilities of the large models and inferring in a zero-shot manner.

4.2 Fine-Tuning & Hyperparameters

All experiments related to transformer-based classifier systems (Fig. 1) were conducted using a single A100 40GB GPU, with the following hyperparameters: 5 training epochs and a learning rate of $2e-5$. Additionally weight-decay and early-stopping (patience = 3) were applied.

For the generative-based systems different environments were used:

- Local setup (Apple M2) + OpenAI access for GPT-4, GPT-3.5-turbo
- Local setup (Apple M2) including Ollama[6] for running Mistral-7b and Llama2

We experimented with low temperature hyperparameter values, ranging $0-0.2$, in order to achieve more deterministic results.

4.3 Custom Legal Dataset

For MLM fine-tuning the transformer classifier, a new custom-tailored U.S. Civil Procedure dataset (Ref. 3.2.2) was used. It was collected first by automatically extracting the keywords from each unique explanation+question entry, then manually creating search queries and finally - using the open search API of the Caselaw Access Project[7] downloading relevant cases. The final corpora consists of 1985 different legal texts (cases), sourced by storing each 20 most relevant results for 100 queries.

4.4 Legal Prompt Engineering (LPE)

In (Trautmann et al., 2022), the authors define "Legal prompt engineering (LPE)" as the process of creating, evaluating, and recommending prompts for legal NLP tasks. In the current work as an alternative approach to the transformer-based classifier systems we investigate the performance of LPE on the SemEval 2024 Legal task. We used more than 15 prompts (A.1), as for their creation, we followed some of the 26 principles described in (Bsharat et al., 2024). Modification of a prompt version was done after evaluating how certain changes affect the performance.

The general frame of the prompt was in the form of a task or question, for which the model has to answer only with "TRUE" or "FALSE". An interesting observation is that GPT-3/4 and Llama2 almost always follow that restriction and return one of the two desired outputs with very few times returning something slightly different (e.g. different casing or appending punctuation - "false.", "True"). Contrarily, Mistral-7b-instruct always returns the answer with an additional explanation, which led to a more complex post-processing step for that model.

We used LangChain[8] for prompt template processing, model-agnostic interface unification and easy response post-processing.

5 Results

Table 1 shows the results of the different experiments. The evaluation of different models for the SemEval-2024 Task 5 on Legal Argument Reasoning presents interesting observation on how different models and system types, described in the current work perform on the dev and test dataset partitions. Our baseline approaches, Majority and Random, set the initial benchmarks with Macro-F1 scores significantly lower than those achieved by advanced models, underscoring the complexity of the task. The application of transformer models, including those equipped with a Sliding Window

[6] https://ollama.com/

[7] https://case.law/

[8] LLM framework - https://python.langchain.com/

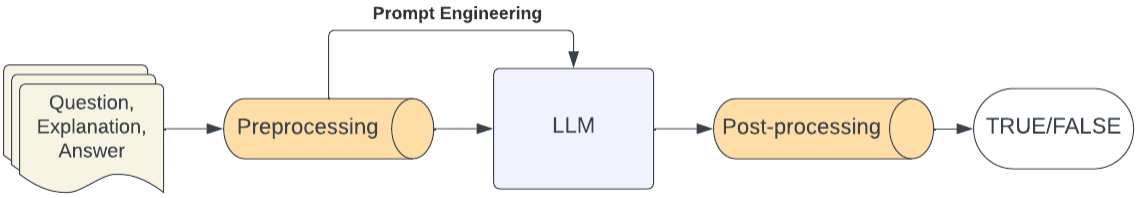


Figure 4: Generative model-based system architecture

Model Name	System Type	Dev Macro-F1	Test Macro-F1
Majority	Baseline	0.44	0.42
Random		0.46	0.46
CaseHold/LegalBERT + SW	Transformer	0.55	-
LegalBERT + SW		0.59	-
LegalBERT-small + SW		0.53	-
InLegalBERT + SW		0.44	-
lexlms/legal-longformer-base		0.50	-
SU-FMI-LegalBERT + SW		0.60	-
lexlms/legal-roberta-large		0.62	0.49
legal-roberta + BART	Classifier + Summary	-	0.50
SU-FMI-LegalBERT + BART		-	0.52
CaseHold/LegalBERT + BART		-	0.54
CaseHold/LegalBERT + GPT-4		-	0.55
CaseHold/LegalBERT + GPT-4 V2		-	0.61
Mistral-7b + LPE	LLM	-	0.58
Llama2-70b + LPE		0.59	0.58
GPT-3.5-turbo + LPE		0.58	0.60
GPT-4 + LPE		0.74	0.7728

Table 1: Model Performance on Development and Test Sets

technique and summarization capabilities, such as BART, showed improvement over the baselines, indicating the value of contextual understanding and content summarization in legal reasoning tasks.

Notably, the integration of Large Language Models (LLMs) with Legal Prompt Engineering (LPE) techniques, particularly with GPT-3.5-turbo and GPT-4, led to a significant leap in performance metrics. These models outperformed traditional transformer models, highlighting the effectiveness of LPE in enhancing the model's ability to interpret and reason over legal texts.

The comparative analysis of model performances on both development and test datasets revealed consistent patterns. Models utilizing LLMs with LPE not only achieved the highest Macro-F1 scores but also demonstrated robustness across different data sets, underscoring their potential for real-world applications in legal reasoning and argumentation.

6 Conclusion

Our participation in SemEval-2024's legal argument reasoning task has yielded valuable insights into the capabilities of transformer-based models and LLMs in processing and reasoning over legal texts. While our methods have shown promise, particularly in leveraging LLMs and prompt engineering, the complexity of legal reasoning poses ongoing challenges.

Further investigation can be done in a solution based on a hierarchical transformer variant such as HIER-BERT (Chalkidis et al., 2022b), (Chalkidis et al., 2019) (Chalkidis et al., 2022a). Our initial experiments with that model architecture did not lead to very high results (0.47 f1-macro on the dev partition) and because of the setup complexity, we decided to leave it for future research opportunities.

7 Acknowledgments

This work was partly supported by the European Union-NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, project No BG-RRP-2.004-0008.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The Legal Argument Reasoning Task in Civil Procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, UAE. Association for Computational Linguistics.

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022a. An exploration of hierarchical attention transformers for efficient long document classification.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Ilias Chalkidis, Nicolas Garneau, Catalina Goanta, Daniel Martin Katz, and Anders Søgaard. 2023. Lex-files and legallama: Facilitating english multinational legal language model development.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022b. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Li Chenxi, Feng Jilin, Huang Meng, and Wang Zhonghao. 2022. Research on post earthquake public opinion analysis based on xlnet-bigru-a algorithm. In *2022 IEEE 2nd International Conference on Computer Communication and Artificial Intelligence (CCAI)*, pages 81–84. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

G. Erkan and D. R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. 2023. Pre-trained language models for the legal domain: A case study on indian law.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases?

Dietrich Trautmann, Alina Petrova, and Frank Schilder. 2022. Legal prompt engineering for multilingual legal judgement prediction.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset.

A Appendix

A.1 Prompts

A.1.1 Best Prompt (GPT-4, GPT-3.5, Llama2-70b

System Prompt:

system_prompt = """

You are a legal assistant with a specialization in U.S. Civil Procedure. Your role involves thorough analysis and resolution of cases pertaining to this field. You will encounter three key components in each case:

1. EXPLANATION: This provides additional context and background information about a specific lawsuit.

2. QUESTION: Here, you will be presented with actual facts and details surrounding the lawsuit.

3. HYPOTHESIS: Based on the provided information, a hypothesis will be presented. Your task is to rigorously evaluate this hypothesis in the context of U.S. Civil Procedure and determine its validity. Respond ONLY with 'TRUE' if you conclude that the hypothesis is correct, or ONLY with 'FALSE' if you find it to be incorrect.

*Do no*t provide any reasoning behind your decision.

"""

User Input:

input_template = """

EXPLANATION: {}

QUESTION: {}

HYPOTHESIS: {}

"""

A.1.2 Chain of thoughts

system_prompt = """

You are a legal assistant with a specialization in U.S. Civil Procedure. Your role involves thorough analysis and resolution of cases pertaining to this field. You will encounter three key components in each case:

1. EXPLANATION: This provides additional context and background information about a specific lawsuit.

2. QUESTION: Here, you will be presented with actual facts and details surrounding the lawsuit.

3. HYPOTHESIS: Based on the provided information, a hypothesis will be presented. Your task is to rigorously evaluate this hypothesis in the context of U.S. Civil Procedure and determine its validity.

On User input with EXPLANATION, QUESTION and HYPOTHESIS analyse the legal problem step by step. Explain your thoughts.

"""

final_input = """

Respond ONLY with 'TRUE' if you conclude that the hypothesis is correct, or ONLY with 'FALSE' if you find it to be incorrect.

Do not provide any reasoning and ONLY answer with 'TRUE' or 'FALSE'

"""

A.1.3 Mistral-7b-instruct Best Prompt

You are a helpful civil law assistant. Your answer only with "TRUE" or "FALSE". You answer with "TRUE" if the STATEMENT is correct based on the provided CONTEXT or "FALSE" otherwise. If you don't know the answer - answer with FALSE.

==========

The CONTEXT is {explanation} | {question}

==========

The STATEMENT is: {answer}

Challenge at SemEval 2024 Task 7: Contrastive Learning Approach on Numeral-Aware Language Generation

Ali Zhunis
University of Tübingen
Tübingen, Germany
ali.zhunis@student.uni-tuebingen.de

Hao-Yun Chuang
National Chengchi University
Taipei, Taiwan
110555010@nccu.edu.tw

Abstract

Although Large Language Model (LLM) excels on generating headline on ROUGE evaluation, it still fails to reason number and generate news article headline with an accurate number. Attending SemEval-2024 Task 7 subtask 3, our team Challenges aims on using contrastive loss to increase the understanding of the number from their different expression, and knows to identify between different number and its respective expression. This system description paper uses T5 and BART as the baseline models, comparing its result with and without the constrative loss. The result shows that BART with contrastive loss have surpassed all the models, and its performance on the number accuracy has the highest performance among all.

1 Introduction

This paper is a description of the methods we have applied for our implementation on this year's SemEval Task 7, NumEval: Numeral-Aware Language Understanding and Generation. SemEval is an annual workshop which is consisted of various natural language processing shared tasks. Teams that join the tasks is required to design systems that could enhance the understanding or improve results on various kinds of semantic evaluation challenge. The task we decide to join was task 7, NumEval: Numeral-Aware Language Understanding and Generation (Huang et al., 2023). Specifically, we focus on the second subtask from the third task in NumEval, which centers on generating proper news headline based on the provided news articles. Different from article summary, a headline must condense the essence from the full length article. Although the encoder-decoder language models nowadays has excelled on generation of the text based on the ROUGE metric, it still fails on providing precise numeral generation in headlines owing to the fact that the representation of the number may differ in various kinds of forms. Therefore, the goal of this task is to enhance the accuracy of the model in the generation of the number from the headline of a news article.

While the numeral expression in the article consists of text and numbers, our system aims to use the technique of contrastive learning. With this technique, it is possible to help the model enclose the similarity between the number and its text expression, and enlarge the difference between different numbers (and its respective expression).

The evaluation of the performance is divided into two parts, one is to evaluate the accuracy of the predicted number, the other is to evaluate the accuracy of the word prediction, which uses ROUGE, BERTScore, and MoverScore to evaluate the results.

Two datasets are included in the task. One is the dry-run dataset that is provided on the official site. It contains a total of 100 instances; the other is the official training set that is provided after the registration of the task, which includes a total of 21157 instances. All of them have the same data structure. Each instance contains both the news article that includes the date of release, and its respective headline. Each team is expected to generate a precise news title from its respective news article.

Numbers are one of the most important element among medical, business, and legal article, and it could be dangerous if the large language model has misunderstood the content of the article. The finding of this paper could further discover a better way for large language model to detect and reason the numbers from the respective article, and thus generate the accurate headline with correct number.

Our system description is divided into three main section. First, we present the main approach that we used as the final submission result in detail, including the preparation of the dataset, its augmentation, and the structure of the model in contrastive learning that we designed in order to solve the task.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1659–1662

Hence, we talk about the adjustment of the model, including parameter optimization on the model. Finally, we present our experimental result. In the end, we discuss about the possible future work, and conclude with a brief summary of our system.

2 Related work

Large Language Model (LLM) like ChatGPT has been long commented with its brittleness on the ability of numerical reasoning. When the questions presented in the varying textual form (comprising words and numbers), LLM would result in inconsistent performance (Ahn et al., 2024). In (Huang et al., 2023)'s work, it shows that although large language model excelled based on ROUGE metrics, it still fails to generate precise numeral in headline.

Researchers have applied contrastive learning in natural language processing. To generate headline with different author style, (Liu et al., 2022) has applied contrastive learning to integrate the stylistic feature of the author into the model. This research hence inspires us to use contrastive learning on integrating the numeral features in different text form to the model, in order to let model identify the correct number from the news article.

3 Methodology

In this section, we propose the methodology of our system. It mainly consists of three parts. First, we will talk about the augmentation of the data. Secondly, the model training and fine-tuning, and finally, parameter optimization.

3.1 Data pre-processing and augmentation

A total of two vectors are used in our experiment. The first vector enhance the model understanding of different expression on number. One is to change all the text expression of number into numeral expression, and vice versa. While it encloses the similarity between the different expression of the same number, we call it positive distorted sample. With this change, model can better learn the different expression from the same number. We manually annotated the dataset to change the number into different kind of form. For example, if the number is *1000*, then it would be transferred into *1K*. The purpose of this is to increase the range of the understanding of the number in all forms.

The other vector, on the other hand, serves as the role that teaches the model to identify different numbers. It helps to enlarge the difference between the different numeric expressions, we call it negative distorted sample. Therefore, we also apply ChatGPT with the prompt[1] to change the number from every news article.

Positively Distorted
30K Walmart Part-Timers to Lose Health Insurance.
Thirty thousand Walmart part-timers to Lose Health Insurance
Negatively Distorted
Dax Shepard: Wedding to Kristen Bell Cost $142.
Dax Shepard: Wedding to Kristen Bell Cost eight hundred

Table 1: Examples of positively and negatively distorted headlines

3.2 Encoder

Transformer Seq2Seq Model (Vaswani et al., 2017) revolutionized the field of sequence-to-sequence learning. The implementation of the self-attention mechanism allowed weighting of the importance of different input tokens during the generation of each output token. The creation of multi-head attention enhanced the ability of the model to capture the diverse relationship between tokens. Based on transformer model advantages, the pre-trained BART-base model (Lewis et al., 2020) was selected as an encoder for headline representation creation. For the comparison, the T5 (Raffel et al., 2020) model was also utilized.

3.3 Models

In this section, we aim to delineate the types of models we trained and the portion of data utilized for training. The first model, referred to as **BART(sub)**, represents the outcomes of a model submitted for evaluation. In this model, the hyperparameters of the pre-trained BART model with Contrastive Learning (CL) were fine-tuned. Following the submission, our focus shifted towards improving and adjusting the Contrastive Learning approach. The subsequent model, named **BART with CL**, was trained using improved contrastive learning techniques. However, due to resource constraints, it was trained solely on **1000** instances of data.

3.4 Contrastive Loss

Our proposed Contrastive Learning enhanced model was implemented on End-to-End Seq2Seq

[1]You are the examiner. Examine the text. If the number in any form appears in the text, change the number into another number. Return the revised text only.

generation model. For the implementation of CL, positive and negative samples of headlines were explicitly constructed. Given news and headline, we trained End-to-End Seq2Seq models to generate headlines based on ground truth headlines. For the CL part, the model uses news, 2 positive and 2 negative headline samples. Given that our main task is specifically numerical aware headline generation, the sampling method was chosen to put more attention to numbers. That is why during the model training, the samples for the batch were exclusively formed from the 2 positive and 2 negative distortions of the same headline. This encourages the model to preserve the semantic content of the headline while allowing variations in numerical values. By explicitly focusing on numerical distortions in the loss function, the model learns to generate headlines that are robust to variations in numerical values. The loss function of the positive pair of examples (i, j) is defined as:

$$\mathrm{L}_s = -\lambda \log \frac{\exp(\tau^{-1} sim(z_i, z_j))}{\sum_s I(s \neq i) \exp(\tau^{-1} sim(z_i, z_j))})$$

where: $I(\cdot)$ is an indicator function such that $I(s \neq i) = 1$ and $I(s = i) = 0$, and τ is a temperature parameter and (s) is an index variable representing the current sample being considered during the training process.

This loss function penalizes the model if the distance between the news and positive headline embeddings is not closer than the distance between the news and negative headline embeddings by a certain margin. The final loss function for the numeric headline generation task will consist of a combination of model loss L_{model} and contrastive loss L_{CL} where the β is hyperparameter. Model Loss is the loss calculated from the model's forward pass using the ground truth labels.

$$Loss = L_{model} + \beta \times L_{CL}$$

3.5 Evaluation Metrics

For the evaluation of trained models, the automatic evaluation metric that the task organizer proposed was utilized. It consists of the ROUGE metric, incorporating ROUGE-1, ROUGE-2, and sentence-level ROUGE-L. For the BERTScore it incorporates BERT Precision, BERT Recall, and BERT F1. Also, the overall, copy and reasoning numerical accuracies were calculated.

3.6 Implementation and Hyperparameters

Due to resource constraints, all our models were trained only on **1000** instances of training data. For each model, we established the maximum length for both the article and the target headline, with values set at 512 and 16 respectively. In our trials, we adapt BART-large, T5-large and our newly introduced CL-augmented model. When tackling the headline generation task, we utilize beam search with a beam size of 8 and configure our batch size to 4. For the contrastive loss, the margin was set as 0.5, and β of 0.5 was selected. We apply the Adam optimizer with a learning rate of 5×10^{-6}. The models undergo training for 10 epochs, with the validation set used to assess performance. All experiments were conducted on the Kaggle T4 GPU.

4 Experimental Results

Table 1 shows the performance from different headline generation models evaluated by ROUGE score. BART model trained with contrastive loss has achieved 35.74, which is higher than other baseline models, showing its effectiveness on headline generation. Comparing BART with and without contrastive loss (CL), we observe a notable improvement in ROUGE scores when contrastive loss is incorporated during training. BART with CL achieves the highest ROUGE-1 at 40.91 and ROUGE-2 at 17.49 scores. Results indicate that contrastive loss regularization enhances the model's ability to generate headlines with higher lexical overlap and coverage.

Table 2 is the BERTScore for each headline generation model. Its BERT F1 score of 41.77 reflects strong semantic similarity to reference summaries, indicating robust performance across both lexical and semantic dimensions.

Model	Rouge-1	Rouge-2	Rouge-L
BART	38.48	15.18	33.35
T5 with CL	36.72	14.31	32.58
BART(sub)	31.22	12.23	26.86
BART with CL	**40.91**	**17.49**	**35.74**

Table 2: ROUGE scores of Headline Generation Models

Model	P	R	F1
BART	33.4	45.61	33.46
T5 with CL	35.50	39.90	37.72
BART(sub)	19.53	**47.56**	33.13
BART with CL	**36.91**	46.67	**41.77**

Table 3: BERT scores of Headline Generation Models

Model	Overall	Copy	Reasoning
BART with CL	72.956	82.170	56.176

Table 4: Numerical accuracy evaluation results

Additionally, BART with CL exhibits a BERT Precision of 36.91 and a BERT Recall of 46.67, further emphasizing its balanced performance in capturing semantic content accurately.

5 Conclusion

Resolving Task 7 at SemEval-2024 as team Challenges, we applied contrastive learning techniques on several models, in order to see which obtain the model with highest performance. In the final submission we obtained the highest number accuracy in the COPY category, up to 82.170, and got the second place in overall score, also up to 72.956. In our human evaluation process, our headline generation model achieved the third-highest level of numerical accuracy by reaching 1.70 score. It thus proves that our approaches can help train the model in numerical reasoning and numerical headline generation. However, this model is merely trained on the **small part of dataset**. Therefore, in future work, it is suggested that more instances might help all the conventions. If we enlarge the data size, it is possible that the performance may get higher. Augmenting positive and negative headline samples artificially may enhance the effectiveness of numeric-based headline generation. In our future work, we are planning to generate positive and negative samples for the whole dataset, and train models.

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Hui Liu, Weidong Guo, Yige Chen, and Xiangyang Li. 2022. Contrastive learning enhanced author-style headline generation.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Team Bolaca at SemEval-2024 Task 6: Sentence-transformers are all you need

Béla Linus Rösener
Student/Uni-Tuebingen
bela.roesener@gmail.com

Ilinca Vandici
Student/Uni-Tuebingen
ilinca.vandici@uni-tuebingen.de

Hong-Bo Wei
Student/Uni-Tuebingen
hong-bo.wei@student.uni-tuebingen.de

Abstract

The prevalence of fluent over-generation hallucinations, grammatically correct but nonsensical text, poses a significant challenge to the reliability of Natural Language Processing (NLP) systems. These fabricated constructs, arising from factors like overfitting or data sparsity, can mislead users and undermine system efficacy. The SemEval-2024 Task 6, SHROOM, addresses this concern by offering a comprehensive evaluation platform. For our own contribution to the task we make use of a logistic regression classifier and a feed-forward ANN in order to provide a computationally economical, yet reliable solution to the the Task at hand.

1 Introduction

Fluent over-generation hallucinations, grammatically correct yet factually incorrect or contextually irrelevant text outputs, remain a persistent obstacle in NLP systems, particularly large language models (LLMs). Moreover, the coherent aspect of the output means that hallucinations are harder to detect than other types of erroneous generation,as discussed in (Guerreiro et al., 2022), particularly in tasks like machine translation, especially considering most metrics for measuring performance only account for fluency rather than correctness (Guerreiro et al., 2022) . In order to ensure that tools like LLMs, which are becoming increasingly popular among the general population, provide the user base with information that is faithful and coherent in the context of various language tasks, research in identifying instances of hallucinations has become necessary. In this paper, we present our contribution for the SemEval 2024 task [1], SHROOM, where we work on solutions for detecting and categorizing hallucinations, using the data made available for different types of language generation tasks, stemming from a model-aware and a model-agnostic track. Additionally, taking into account the fact that earlier studies have adopted a LLM-based few or zero shot learning approach to the problem, we opt for a computationally economical approach instead, using a two-pronged model making use of logistic regression and a simple feed-forward network.

2 Task Description

SHROOM challenges participants to develop a model-agnostic or model-aware binary classification system capable of identifying fluent overgeneration hallucinations in diverse NLP tasks like definition modeling, machine translation, and paraphrase generation.

The data consists of 61,080 text outputs, of which 1,080 are manually annotated instances (the rest being unlabeled).

3 Background

SHROOM represents a pivotal benchmark for advancing NLP systems' ability to discern and categorize fluent over-generation hallucinations. Its focus on real-world applicability through the model-agnostic track and its diverse dataset empower researchers to assess the limitations and strengths of current techniques. Ultimately, SHROOM contributes to the broader mission of enhancing the trustworthiness and resilience of NLP systems, a crucial aspect for applications like machine translation, text summarization, and chatbot interactions.

The task requires participants to develop a binary classification system which successfully identifies hallucinations for different types of language generation tasks: definition modeling, machine translation and paraphrase generation. The data was generated from two different tracks, model-aware, meaning knowlegde of the model which produced

[1]Our code is available for replication purposes at https://github.com/cicl-iscl/SemEval2024$_T$6$_S$$HROOMS$

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1663–1666

the output is accessible, and model-agnostic, where the model which generated the ouput is unknown. The generated outputs are provided in JSON format, containing source text, generated text and golden standard prompt, aswell the model name when applicable. 61080 datapoints are obtained in this way, from which 1080 are annotated (Mickus et al., 2024). A baseline was made available, using a zero-shot model with calls to LLAMA (Mickus et al., 2024).

4 Our System Strategy

Considering the likely instances where a system to detect hallucinations would find practical use and how current approaches to hallucination detection work (Friel and Sanyal, 2023), we decided that we wanted to make reduced inference time a goal of our system. Our primary strategy for detecting hallucinations involves a two-pronged approach utilizing either logistic regression or a small feed-forward neural networks trained on a labeled dataset as classification model. This leverages the strengths of each model for efficient and accurate hallucination detection. As input to our classification model we use sentence embeddings generated by SBERT (Reimers and Gurevych, 2019).

Using a logistic regression model as a baseline helps us to quantify the advantages of using a neural network as a classification model instead of simpler approaches.

4.0.1 SBERT

In order to enrich our understanding of the text and capture deeper semantic relationships beyond surface-level similarities, we incorporate Sentence-BERT (SBERT) embeddings into our system. SBERT generates high-dimensional vector representations of text, encoding semantic meaning and context.

We obtain reliable vector representations of our data utilizing a pre-trained SBERT model (e.g., all-mpnet-base-v2, all-MiniLM-L6-v2), we generate vector representations for each the source, target and hypothesis fields of our inputs. These vectors can be envisioned as high-dimensional fingerprints capturing the semantic essence of each sentence and its relationship to others. These SBERT-derived features are integrated with features such as task and model. This enriched feature set provides a comprehensive representation of the text, capturing deeper semantic information.

SBERT enables us to transcend basic word-level comparisons, allowing us to capture meaning and context within text more comprehensively. SBERT takes into account the context surrounding each sentence during analysis, which aids in identifying variations from the intended meaning and inconsistencies within the text. We anticipate that combining traditional features with those derived from SBERT will enhance the accuracy and generalizability of hallucination detection.

Additionally, using SBERT fits into our lightweight approach to the task, by offering a fast tool for inference, being more lightweight than newer state-of-the-art models.

4.0.2 Logistic Regression

For the initial layer of analysis, we employ logistic regression as a robust baseline model. Its interpretability allows us to gain insights into the key features distinguishing genuine and hallucinated text. We use SBERT to encode a prompt that incorporates Source, Target and Hypothesis. This provides the logistic regression model with single vector as input. The logistic regression model is then trained on these features to learn the underlying patterns that differentiate hallucinated and non-hallucinated text. This simple method provided us with a simple baseline.

4.0.3 Artificial Neural Network

The classification network is a simple multilayer feed-forward network. Its input are three sentence embeddings generated by SBERT from the Source, Target and Hypothesis fields of the input, as well as other features that the input provides. The usage of a neural network allows us to capture non-linear relationships and hidden patterns within the data that might be missed by the logistic regression model. The ANN architecture is designed with multiple hidden layers and non-linear activation functions, enabling it to learn intricate feature interactions and representations.

4.1 Key Discoveries and Challenges

The task presented us with two small datasets containing labeled instances and a bigger dataset without labels. Our main challenge therefore was to make the best use of a very limited dataset or find ways to leverage the not annotated data.

In regards to the following step in our approach, namely feature extraction, while many of the instances contained in the dataset seem to require

deep semantic analysis of the input data, much simpler features of the input can also be useful to identify hallucinations: word repetitions, n-gram counts, output length, unexpected characters, etc. (Huang et al., 2023)

Pertaining to supervised learning, while the approach proved itself to be useful, it also revealed its limitations. The reliance on pre-labeled data can restrict the generalizability of the model to new domains or tasks. Additionally, the quality of the pre-labeled data can impact the model's performance.

One of the main challenges we encountered was the lack of labeled data. Classifying hallucinations in unlabeled data remains a challenging task. The absence of explicit labels hinders the model's ability to definitively determine whether an instance is a hallucination. This problem highlights the need for more sophisticated methods for dealing with unlabeled data. However, the inclusion of the pre-labeled data which was made available for training our hallucination detection model proved to be an effective strategy. The model was able to generalize well to unseen data and achieve significant accuracy in identifying hallucinations, indicating that access to annotated datasets documenting cases of over generation will of course improve classification.

Our exploration of the dataset revealed that over-generation, the production of excessive or irrelevant text, is a significant aspect of hallucinations. The ability to distinguish between fluent over-generation and genuine hallucinations poses an additional challenge for our system, and is an aspect which would need to be further explored.

5 Key Algorithms and Modeling Decisions in Our SHROOM System

Our hallucination detection system employs a hybrid approach that combines both a pretrained LLM and a small classification model. The system's core components include:

1. Pre-trained Language Model (LM): We utilize a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model to extract linguistic features from the text outputs ((Reimers and Gurevych, 2020)). BERT's ability to capture contextual information and semantic relationships is crucial for understanding the nuances of language and identifying deviations from the intended meaning. We use SBert for its ability to provide meaningful sentence embeddings and while being faster and less resource intensive than the newest LLMs openly available. This would make running our model in practical applications more realistic.

2. Classification Model: The sentence embeddings produced by the BERT-Model are given to 1) a logistic regression model or 2) a small feed-forward network producing a label probability.

3. Supervised Learning from Hallucination Annotations: The feed-forward network is trained using the labeled data provided in the SHROOM dataset. The model learns to classify text outputs as either containing hallucinations or being truthful to the Source.

6 Results

We achieved an accuracy of 0.57 on the model-aware track, and an accuracy of 0.63 on the model-agnostic track, placing us at respectively rank 32 and rank 27 on the competition leaderboard. Additionally, we scored 0.24 for accuracy for the model-agnostic track.

7 Experimental Setup

Because of our approach based on supervised learning we used the development dataset provided by the task organizers as training dataset, using cross validation to gain insights into our systems' performance before the actual test dataset was available.

We used PyTorch: 1.10.2 [2] in order to build our neural network. Transformers 4.12.2 [3], more specifically SBERT [4], was used in order to extract the sentence embeddings. Finally, in order to organize and process the data, we also made use of NumPy (1.22.3) [5] and Pandas (1.4.2)[6]. In order to run our code in an efficient manner, we used Colab[7].

8 Conclusion

By using a simple model exploiting feature extraction to aid in identifying hallucinations in a dataset containing data from different tasks, we

[2]https://pytorch.org/
[3]https://github.com/huggingface/transformers/
[4]https://www.sbert.net/
[5]https://numpy.org/
[6]https://pandas.pydata.org/
[7]https://colab.research.google.com/

have achieved an accuracy of 0.628. This could potentially indicate that for this type of task, it might be worthwhile to take into consideration approaches which do not exclusively rely on zero-shot classification, but instead make use of less computationally costly techniques. We have shown that such methods are not only efficient, but also present the advantage of being easily reproducible with fewer resources.

9 Going Forward

While our system is, at its current state, not usable in production systems, it shows that computationally less expensive methods can still lead to working systems in a task as complex as hallucination detection. Our implementation still leaves room for improvement and some unexplored possibilities: Our system does not leverage the unlabeled training data provided with the task. Using an encoder-decoder architecture to pretrain an encoder layer for the classification model might improve its training results on the small labeled data set. SBERT embeddings can be used to detect meaning similarities of texts, adding combinations of different embeddings (Source+Hypothesis, Target+Hypothesis, ...) may provide useful features to the classification layer. Our aim for a lightweight and fast system also makes manual approaches of hallucination detection as described in (Bruno et al., 2023) attractive.

References

Alessandro Bruno, Pier Luigi Mazzeo, Aladine Chetouani, Marouane Tliba, and Mohamed Amine Kerkouri. 2023. Insights into classifying and mitigating llms' hallucinations. *arXiv preprint arXiv:2311.08117*.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *arXiv preprint arXiv:2310.18344*.

Nuno M Guerreiro, Elena Voita, and André FT Martins. 2022. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. *arXiv preprint arXiv:2208.05309*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

AIpom at SemEval-2024 Task 8: Detecting AI-produced Outputs in M4

Alexander Shirnin [1], **Nikita Andreev** [2]
Vladislav Mikhailov [3], **Ekaterina Artemova** [4]
[1]HSE University, [2]CAIT and Applied AI Institute
[3]University of Oslo, [4]Toloka AI
Correspondence: ashirnin@hse.ru

Abstract

This paper describes AIpom, a system designed to detect a boundary between human-written and machine-generated text (SemEval-2024 Task 8, Subtask C: Human-Machine Mixed Text Detection). We propose a two-stage pipeline combining predictions from an instruction-tuned decoder-only model and encoder-only sequence taggers. AIpom is ranked second on the leaderboard while achieving a Mean Absolute Error of 15.94. Ablation studies confirm the benefits of pipelining encoder and decoder models, particularly in terms of improved performance.

1 Introduction

SemEval-2024 Task 8 (Wang et al., 2024a) focuses on multigenerator, multidomain, and multilingual machine-generated text detection based on the M4 corpus (Wang et al., 2024b). The shared task offers three subtasks, which correspond to standard task formulations in the rapidly developing field of artificial text detection (Jawahar et al., 2020; Uchendu, 2023): (A) classifying if a given text in a particular language is human-written or machine-generated, (B) attributing the author of a given text, and (C) detecting a boundary between human-written and machine-generated text. Developing generalizable solutions to these problems helps mitigate the risks of misusing generative language models (LMs) for malicious purposes (Weidinger et al., 2022) and improve human performance in identifying AI-produced content (Gehrmann et al., 2019).

This paper proposes AIpom[1], a novel method for human-machine mixed text detection (Subtask C). The boundary detection setup aligns with common user scenarios for applying generative LMs in practice, e.g., text continuation, creative writing, and story generation. The standard approach to this task is training a linear classifier or a regression model over encoder representations (Cutler et al., 2021; Dugan et al., 2023). In contrast, AIpom leverages a pipeline of decoder and encoder models to detect machine-generated text, utilizing them sequentially. AIpom takes second out of 33 participating teams on the Subtask C leaderboard by achieving a Mean Absolute Error (MAE) of 15.94 on the official evaluation set. After the official evaluation phase, we develop a better-performing solution with an MAE score of 15.21.

Our ablation studies confirm that using decoder or encoder models individually leads to lower performance. Thus, employing the pipeline of decoder and encoder models proves to be an effective solution. Additionally, these studies highlight domain shift issues, as there is a significant score disparity between the development and official evaluation sets. Future efforts should focus on enhancing the AIpom robustness with respect to the text domain and text generator. The codebase and models are publicly released[2].

2 Background

The M4 corpus consists of human-written and machine-generated texts in six languages (English, Chinese, Russian, Urdu, Indonesian, and Arabic) across various domains, ranging from Wikipedia to academic peer reviews. The generative LMs include the OpenAI models (ChatGPT and text-davinci-003), LLaMA-1 (Touvron et al., 2023), FLAN-T5 (Chung et al., 2022), Cohere, Dolly-v2[3], and BLOOMZ (Muennighoff et al., 2022). The organizers provide 3649, 505, and 11123 dataset instances in Subtask C training, development, and official evaluation sets, respectively.

[1]AIpom is named after a simian pokémon *aipom*, and stands for detecting **AI**-**p**roduced **o**utputs in **M**4.

[2]github.com/25icecreamflavors/AIpom
[3]hf.co/databricks/dolly-v2-12b

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1667–1672

Task Formulation Human-machine mixed text detection requires predicting the index corresponding to the first machine-generated word, as shown below:

- `text:` "👨‍💼We have added a 2+ page 🤖discussion on the experimental results, highlighting the superiority of the ARC-based models and their impact on the field of deep learning."
- `label: 6`

Performance Metric MAE measures the absolute distance between the predicted word and the word where the human-machine transition occurs.

3 AIpom

First, we overview the AIpom pipeline. Next, we detail the fine-tuning procedures for encoder and decoder models.

Overview The AIpom pipeline (see Figure 1) consists of multiple consecutive steps of fine-tuning language models:

(a) The decoder is fine-tuned on the training set to predict the change point from a human-written text to a machine-generated text.
(b) The decoder makes predictions and outputs the source texts with predicted change points.
(c) The first encoder model is fine-tuned on the texts with predicted change points from step (b).
(d) The second encoder model is fine-tuned on the mixture of texts from the training set and the texts with predicted change points from step (b).
(e) Two encoders are used to predict the indices of change points in test texts.
(f) The predicted change points from step (e) are aggregated by averaging.

Decoder The decoder is fine-tuned as follows: the input comprises the prompt and the training text. We experimented with various prompts, including instructing the model to output only the human-written text, the text with an inserted symbol representing the change point, and the machine-generated text alone. Our preliminary experiments suggest that instructing the decoder to output only the machine-generated text yields better results. Therefore, we use this option in subsequent experiments. Table 1 describes the prompt, and Figure 2 illustrates fine-tuning the decoder. The decoder is used in the first step of the AIpom pipeline: we utilize it to generate initial predictions, which are then further processed by two encoders.

```
As an output, write only the machine-generated part of the provided text. Output must start with "Answer: ". Separate tokens by " ". If the whole text is human-written, output "None". Here is the text: example["text"]
```

Table 1: The prompt used for fine-tuning the decoder.

After receiving the predicted text from the decoder, we post-process the original training text and insert a special token <BREAK> directly before the first machine-generated word predicted by the decoder. This allows us to pass the prediction further to the encoder.

Encoder The encoder is fine-tuned to label input texts on a token-wise way. Each token in the human-written segment is labeled with a zero, while each machine-generated token is labeled with one. In our final prediction, we determine the position of the word in which the first "1" label appears, indicating machine-generated text. See Figure 3 for illustration.

The AIpom pipeline involves fine-tuning two encoders. The first encoder is trained on a dataset consisting of texts labeled by the decoder. In contrast, the second encoder is fine-tuned using a dataset that includes both the decoder's predictions and the original source texts from the training set. we receive the predicted change point from each encoder, which we aggregate by averaging.

4 Experiments

Overview We design a series of experiments using two recent language models, a decoder *Mistral-7B-OpenOrca*[4] (Jiang et al., 2023) and an encoder *DeBERTaV3-Large*[5] (He et al., 2023), selected based on their performance on standard NLP benchmarks and computational requirements.

First, we establish a baseline for the decoder model by zero-shot prompting and then compare it to Low-Rank Adaptation (LoRA) tuning (Hu et al., 2021), which yields significantly better results. Second, we look into improving the performance of the

[4] hf.co/Open-Orca/Mistral-7B-OpenOrca
[5] hf.co/microsoft/deberta-v3-large

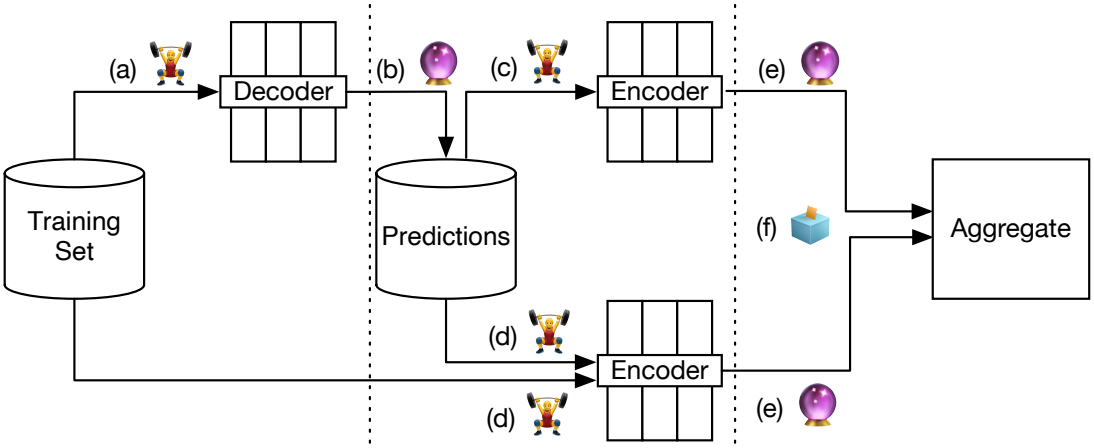


Figure 1: The AIpom pipeline involves fine-tuning decoder and encoder models to predict change points between the human-written and machine-generated text. This process includes fine-tuning the decoder, predicting change points, fine-tuning two encoders, and aggregating predicted change points. 🏋 stands for fine-tuning a language model, 🔮 – predicting with the language model, 📦 – for aggregating the predictions by averaging.

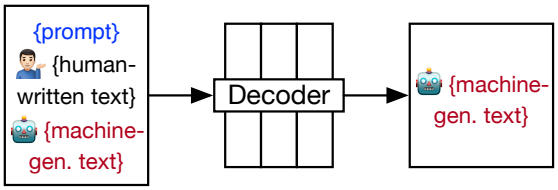


Figure 2: We fine-tune the decoder to output only the machine-written text.

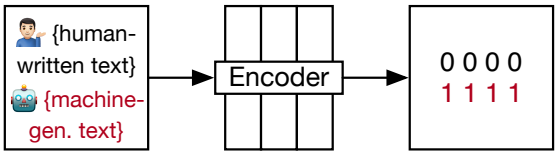


Figure 3: We fine-tune the encoder for token labeling. Human-written tokens are assigned zeros, while machine-generated tokens are assigned ones.

encoder model. We experiment with hyperparameter selection and feeding the encoder model with texts labeled by the decoder model. The combination of the decoder and encoder model outperforms each pipeline component individually.

We only use the development set to evaluate our pipeline and choose our final submission based on the MAE on the development set. In §5, we report the ablation studies results on both development and official test sets[6].

Decoder fine-tuning and inference To fine-tune the Mistral model, we employ LoRA layers tuning with the `SFTTrainer` class from the transformers library (Wolf et al., 2020). The model is fine-tuned on to output machine-generated texts, that is the loss functions are computed only on the model-generated parts. We experimented with learning rates in the range $[1e\text{-}5, 5e\text{-}5]$ with an increment of `1e-5`, and `warmup_ratio` $\in \{0.01, 0.03, 0.05\}$. Based on the results observed on the development set, we select a learning rate of `2e-5`, combined with a `warmup_ratio=0.03` and the `CosineLRScheduler`. For the Parameter-Efficient Fine-Tuning (PEFT) configuration, we adhere to the recommended parameters for Mistral: `rank=32`, `lora_alpha=64`, and `lora_dropout=0.05`. The batch size is set to 4 and the model is fine-tuned for 4 epochs.

We fine-tune the model to start its response with the `"Answer: "` template. This helps improve performance at the inference stage by providing easier-to-clean-up predictions, ensuring they always start the same way. We use the vLLM framework[7] (Kwon et al., 2023) for text generation, with default hyperparameters, the sampling temperature of 1, and `top_p` of 1.

Data labeling with decoder To prepare the training set for the encoder, we split the training set into two folds and perform LoRA tuning on two Mistral models with the same hyperparameters on each fold. Then, each fold is labeled using the decoder fine-tuned on the other fold. This helps us track the decoder's performance and reduce overfitting. During testing, we fine-tune another Mistral model on the entire training set and assess its per-

[6]The shared task organizers have released the gold annotation for the official test set.

[7]`github.com/vllm-project/vllm`

Setup	Model	Fine-tuning setup	<BREAK> in the input	Dev MAE	Test MAE
1.	LoRA Mistral	Training set		2.41	17.00
2.	DeBERTa	Pred. from Mistral	✓	1.74	17.15
3.	DeBERTa	Training set + pred. from Mistral	✓	1.74	**15.21**
4.	AIpom	2. + 3.	✓	**1.68**	15.94
		Ablation experiments			
5.	zero-shot Mistral	Training set		56.51	80.81
6.	DeBERTa	Training set		2.15	19.97
7.	DeBERTa	Training set + pred. from Mistral		1.91	16.49

Table 2: MAE scores on the development and official test sets for different setups and ablation experiments. Setup details include the model used, fine-tuning setup, and presence of <BREAK> in the input data at the inference stage. The top table shows each language model's performance in the AIpom pipeline. The bottom part shows ablation experiments.

formance on the development set. It is worth noting that we apply the post-processing step described in §3, specifically in the "Decoder" paragraph, to the predicted text before passing it to the encoder.

Encoder fine-tuning We build upon the baseline code provided by the task organizers[8], enhancing it to effectively fine-tuning the DeBERTa model. We explore a range of learning rates [1e-5, 5e-5] with an increment of 1e-5 to identify the optimal value for fine-tuning our model. Our final fine-tuning strategy utilizes the Adam optimizer (Kingma and Ba, 2015), with a learning rate of 3e-5 and the default CosineLRScheduler. To ensure consistency across all experiments, we use a maximum sequence length of 1024 for text tokenization, maintain a constant batch size of 64, and limit the maximum number of epochs to 6. To reduce overfitting, we freeze a certain number of bottom DeBERTa layers. Specifically, we experiment with fine-tuning only the top $N \in \{6, 12, 18\}$ layers out of the total 24. Through experiments, we determine that fine-tuning only the top 12 layers produces the best results.

Hardware specification We run experiments on a single GPU TESLA A100 80 GB. Model fine-tuning is conducted using the transformers library. The fine-tuning for DeBERTa requires approximately 3.5 hours to complete, while the inference on the official test set runs within 15 minutes. The LoRA tuning for Mistral lasts approximately 12 hours, with the inference on the official test set taking a few hours. To speed up the prediction phase, we employ the vLLM framework, designed specifically for optimizing the inference. This implementation significantly reduces inference time, with the official test set predictions generated in just 30 minutes.

5 Results

Table 2 provides a detailed comparison of the performance metrics across different models and experimental setups. The key results are:

- Fine-tuning Mistral with LoRA tuning (setup 1) on the training set outperforms zero-shot prompting (setup 5) by a wide margin.
- Fine-tuning DeBERTa using the Mistral predictions (setup 2) leads to higher results than fine-tuning DeBERTa on the training set (setup 6).
- Adding a <BREAK> token to the input at the inference stage improves the performance of the DeBERTa model (setup 3 vs. setup 7).
- Averaging predictions of two DeBERTa models (setup 4) leads to the best results on the development set.

The overall best results on the official test set are achieved with setup 3, where the DeBERTa model is fine-tuned on a dataset consisting of both the training set and predictions from the Mistral model, and the <BREAK> token is added to the input at the inference stage.[9]

Decoder vs. encoder Fine-tuned encoder models exhibit inferior performance compared to LoRA-tuned decoder. Specifically, the decoder struggles to comprehend the task in a zero-shot setting, evidenced by a high MAE of 80.81 on the official test set. However, with LoRA tuning, the decoder achieves a significantly lower MAE of 17.00, outperforming the single encoder model's MAE of

[8] github.com/mbzuai-nlp/SemEval2024-task8

[9] These results are achieved after the shared task submission deadline and hence not submitted for the official evaluation stage.

19.97. The encoder models are adaptable and integrate diverse inputs, including prompts and predictions from additional decoder models.

Benefits of pipelining We hypothesize that encoder models benefit from integrating inputs from a decoder. Our pipeline yields the final MAE of 15.94 while fine-tuning only the single encoder model results in a higher MAE score of 19.97.

Robustness While averaging predictions helps improve the overall performance, we find that the AIPom's robustness has room for improvement. In particular, we observe the performance decrease when comparing the results on the development and official test sets. Setup 3, with an MAE of 1.74 on the development set, performs better than an MAE of 15.21 on the official test set. At the same time, Setup 4 (our final submission) achieves a slightly better development MAE but a worse MAE on the official test set. Setup 3 involves finetuning on a mixture of data, showing how using more data can boost the performance and improve the robustness, especially when the dataset is small. We leave improving the out-of-domain robustness for future work.

6 Conclusion

This paper presents the AIpom system submitted to SemEval-2024 Task 8. Our solution achieves 2nd place out of 33 participating teams in Subtask C. We introduce a novel method that utilizes a pipeline of decoder and encoder models. The advantage of this approach is that the models are exposed to both the original data and the predictions from previous steps. We believe this approach holds significant potential, as it allows for creating pipelines comprising various models, mimicking the transfer of learned knowledge. We plan to further improve our system by exploring different combinations of models and longer pipelines. Additionally, we aim to enhance the system's robustness to handle domain-shift scenarios.

Acknowledgements

AS's work results from a research project implemented in the Basic Research Program at the National Research University Higher School of Economics (HSE University). We acknowledge the computational resources of HSE University's HPC facilities.

References

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling Instruction-Finetuned Language Models. *arXiv preprint arXiv:2210.11416*.

Joseph Cutler, Liam Dugan, Shreya Havaldar, and Adam Stein. 2021. Automatic Detection of Hybrid Human-Machine Text Boundaries.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or Fake Text?: Investigating Human Ability to Detect Boundaries between Human-written and Machine-Generated Text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12763–12771.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual Generalization through Multitask Finetuning. *arXiv preprint arXiv:2211.01786*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models.

Adaku Uchendu. 2023. *Reverse Turing Test in the Age of Deepfake Texts*. Ph.D. thesis, The Pennsylvania State University.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, et al. 2024a. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation, SemEval*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. 2022. Taxonomy of Risks Posed by Language Models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

CLaC at SemEval-2024 Task 2: Faithful Clinical Trial Inference

Jennifer Marks[1] **MohammadReza Davari**[1,2] **Leila Kosseim**[1]

[1] Concordia University [2] Mila – Quebec AI Institute

jennifer.marks@mail.concordia.ca
{mohammadreza.davari,leila.kosseim}@concordia.ca

Abstract

This paper presents the methodology used for our participation in SemEval 2024 Task 2 (Jullien et al., 2024) – *Safe Biomedical Natural Language Inference for Clinical Trials*. The task involved Natural Language Inference (NLI) on clinical trial data, where statements were provided regarding information within Clinical Trial Reports (CTRs). These statements could pertain to a single CTR or compare two CTRs, requiring the identification of the inference relation (entailment vs contradiction) between CTR-statement pairs. Evaluation was based on F1, Faithfulness, and Consistency metrics, with priority given to the latter two by the organizers. Our approach aims to maximize Faithfulness and Consistency, guided by intuitive definitions provided by the organizers, without detailed metric calculations. Experimentally, our approach yielded models achieving maximal Faithfulness (top rank) and average Consistency (mid rank) at the expense of F1 (low rank). Future work will focus on refining our approach to achieve a balance among all three metrics.

1 Introduction

Clinical trials serve as the cornerstone for evaluating the efficacy and safety of novel medical interventions, playing a pivotal role in advancing healthcare practices (Avis et al., 2006). Clinical Trial Reports (CTRs) encapsulate crucial information regarding trial methodologies and outcomes, serving as indispensable resources for healthcare professionals in treatment decision-making (Bastian et al., 2010). However, the sheer volume of available CTRs, coupled with their rapid proliferation, poses significant challenges for comprehensive literature review and evidence synthesis in clinical practice (DeYoung et al., 2020). Natural Language Inference (NLI) emerges as a promising approach to address this issue (Bowman et al., 2015; Devlin et al., 2018; Raffel et al., 2020), facilitating the scalable interpretation and retrieval of medical evidence (Davari et al., 2020; Sutton et al., 2020; Davari et al., 2019). The SemEval 2024 Task 2 (Jullien et al., 2024) on *Safe Biomedical NLI for Clinical Trials* extends this paradigm to enable automated inference of relationships between statements and CTRs, thus streamlining evidence extraction and enhancing decision-making processes in healthcare.

The 2024 task is a continuation of the one introduced by Jullien et al. (2023b,a), specifically it focuses on Track 1, which focuses on NLI in the context of clinical trials. In this task, the input consists of pairs of Clinical Trial Reports (CTRs) and corresponding statements, where the statements make claims about the information contained within the CTRs. The objective is to determine the inference relation between each CTR-statement pair, classifying them as either entailing or contradicting each other. For instance, given a statement "Drug X is effective in treating condition Y" and a CTR outlining a clinical trial testing Drug X's efficacy, the task is to determine whether the statement is entailed by the CTR or contradicted by it. The datasets used are similar to those introduced by Jullien et al. (2023b), and further details can be found in there work.

Our system primarily focuses on maximizing Faithfulness and Consistency in the context of SemEval 2024 Task 2 (Jullien et al., 2024). To achieve this goal, we adopt a strategy centered around introducing controlled input noise during model training. This approach is based on the hypothesis that a certain level of tolerance towards input perturbations could enhance the faithfulness and consistency of the generated models. Specifically, we experiment with randomly masking a percentage ($k\%$) of tokens in both Clinical Trial Reports (CTRs) and the corresponding statements, thereby exposing the model to varying degrees of input uncertainty.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1673–1677

Through these experiments, we aim to optimize model performance in capturing the relationships between statements and CTRs, ultimately improving the system's effectiveness in clinical trial inference tasks.

Through our participation in SemEval 2024 Task 2 (Jullien et al., 2024), we observed a notable trade-off between different evaluation metrics. While our approach successfully improved Faithfulness and Consistency metrics, it came at the expense of F1 scores. This finding underscores the challenge of balancing these evaluation criteria and thus the need for future refinement to achieve a more harmonious optimization across all relevant metrics. Specifically, our models achieved top-ranking levels of Faithfulness but demonstrated only average performance in Consistency metrics, resulting in lower ranks in F1 assessment. See Sec. 4 for details.

2 System Overview

In our system, we leverage BART (Lewis et al., 2019) as the primary model for all experiments due to its robustness and effectiveness in various natural language processing tasks, particularly in Natural Language Inference (NLI) (Lewis et al., 2019; Barker et al., 2021; Farahnak et al., 2020). To streamline the fine-tuning process and enhance efficiency, we adopt the LoRA technique proposed by Hu et al. (2021), which significantly reduces the fine-tuning time without sacrificing performance. Additionally, we incorporate the Contrastive Tension loss function introduced by Carlsson et al. (2020). to guide the fine-tuning process. This loss function promotes contrastive learning by separately encoding the Clinical Trial Reports (CTRs) and their associated statements using two copies of BART (Lewis et al., 2019) during each training instance. By allowing only one copy to update its parameters at a time, the model is encouraged to focus on learning the essential semantic relationships between the CTRs and the statements.

Moreover, we introduce a novel approach to enhance the robustness of the model by incorporating random token masking during each training instance. Specifically, we randomly mask a percentage ($k\%$) of tokens in both the CTRs and their associated statements. This introduces noise in the input data, forcing the model to adapt to varying degrees of input uncertainty and preventing it from relying solely on superficial patterns. The rationale behind this approach is to encourage the model to concentrate on the fundamental semantic content of the input rather than exploiting surface-level correlations.

Balancing between the three required metrics—Faithfulness, Consistency, and F1—proved to be the primary challenge in our experimental setup. While optimizing for one metric often led to improvements in its performance, it frequently came at the expense of others. We explored different strategies to strike a balance between these metrics. However, finding an optimal solution that simultaneously maximized all three metrics remained unsolved. Our system struggled to maintain high levels of F1 score while simultaneously improving Faithfulness and Consistency metrics. Further exploration of optimization strategies and leveraging ensemble methods may offer potential avenues for achieving a better balance between the metrics.

3 Experimental setup

Training Details The training, validation, and test data for our experiments were all provided by the organizers of the SemEval 2024 Task 2, as outlined by Jullien et al. (2024). We trained our model for a total of 40 epochs, employing the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001, with a batch size of 32. To stabilize and accelerate training, we implemented gradient clipping (Zhang et al., 2019) with a maximum norm of 1.

Additionally, we incorporated a linear warmup stage consisting of 40 gradient steps followed by a Cosine Annealing learning rate schedule (Loshchilov and Hutter, 2016). This strategy enabled gradual adjustment of the learning rate during the initial phase of training, allowing the model to converge more smoothly towards an optimal solution.

Furthermore, we limited the maximum sequence length to 256 tokens for both CTRs and their corresponding statements, aligning with the model architecture and computational capabilities. Consequently, each sequence pair was truncated to a total maximum of 512 tokens. This limitation on sequence length helped in handling memory constraints and optimizing the processing of input sequences during training. Exploring larger context sizes could be advantageous for future improvements in the task.

Evaluation Metrics Our system's performance is measured through three key metrics: (1) Faithfulness, (2) Consistency, and (3) F1 score.

Faithfulness: Faithfulness measures the degree to which a given system arrives at the correct prediction for the correct reason. Intuitively, this is estimated by assessing the model's ability to correctly change its predictions when subjected to a semantic altering intervention. Let N denote the number of statements x_i in the contrast set (C), y_i represent their respective original statements, and $f()$ denote the model predictions. Faithfulness is computed using Equation below:

$$\text{Faithfulness} = \frac{1}{N}\sum_{1}^{N} |f(y_i) - f(x_i)| \quad (1)$$

where $x_i \in C$, $\text{Label}(x_i) \neq \text{Label}(y_i)$, and $f(y_i) = \text{Label}(y_i)$.

Consistency: Consistency aims to measure the extent to which a given system produces the same outputs for semantically equivalent problems. It assesses the system's ability to predict the same label for original statements and contrast statements for semantic preserving interventions. Even if the final prediction is incorrect, the representation of the semantic phenomena should be consistent across the statements. Let N denote the number of statements x_i in the contrast set (C), y_i represent their respective original statements, and $f()$ denote the model predictions. Consistency is computed using Equation below:

$$\text{Consistency} = \frac{1}{N}\sum_{1}^{N} 1 - |f(y_i) - f(x_i)| \quad (2)$$

where $x_i \in C$, $\text{Label}(x_i) = \text{Label}(y_i)$.

F1: The F1 score is a commonly used metric in NLP tasks (Yang et al., 2023; Davari, 2020), measuring the balance between precision and recall of a model's predictions. It is calculated based on the geometric mean of precision and recall, where precision represents the ratio of true positive predictions to the total number of predicted positive instances, and recall represents the ratio of true positive predictions to the total number of actual positive instances.

4 Results

Our experimental results demonstrate a clear trade-off between the level of token masking (k) and the performance metrics of F1, Faithfulness, and Consistency. As we increase the masking level, we observe a consistent trend of decreasing F1 scores alongside increasing Faithfulness and Consistency metrics.

For $k = 0$, representing no token masking, we observe an F1 score of 0.65 (ranking 22 out of 32), a Faithfulness score of 0.51 (ranking 21 out of 28), and a Consistency score of 0.54 (ranking 25 out of 30). As we progressively increase k, we observe a gradual change in the metrics, specifically decrease in F1, and increase of the other 2 metrics. At $k = 30\%$, the highest masking level tested, we observe a significant drop in F1 score to 0.06 (ranking 28 out of 31), accompanied by substantial increases in Faithfulness (0.95, ranking 1 out of 28) and Consistency (0.6, ranking 22 out of 32) metrics.

Given the substantial decrease in F1 scores beyond a masking level of $k = 30\%$, we did not explore higher masking levels. This decision was made due to the observed trade-off, where increasing token masking beyond a certain threshold led to disproportionately low F1 scores, potentially indicating a loss of model generalization and predictive performance.

5 Conclusion

Our experiments underscore the intricate balance between token masking levels and performance metrics in biomedical NLI for clinical trials. We observed a discernible trade-off: while increasing token masking improves Faithfulness and Consistency, it results in diminished F1 scores. This finding highlights the necessity of exploring future approaches that could better optimize the model with multiple evaluation criteria as their objective function. Additionally, another potential avenue for improvement would involve examining alternative metrics to provide further insights into the behaviour of the model (Davari et al., 2022a; Farahnak et al., 2021; Steck et al., 2024; Davari et al., 2022b). Based on our findings, one avenue of research involves refining token masking strategies to achieve a more optimal balance between F1, Faithfulness, and Consistency metrics. Furthermore, exploring ensemble methods and alternative fine-tuning strategies could provide valuable insights into enhancing the overall performance of the model.

Acknowledgements

The authors would like to thank the organisers of the SemEval shared task and the anonymous reviewers for their comments on the previous version of this paper. This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Nancy E Avis, Kevin W Smith, Carol L Link, Gabriel N Hortobagyi, and Edgardo Rivera. 2006. Factors associated with participation in breast cancer treatment clinical trials. *Journal of Clinical Oncology*, 24(12):1860–1867.

Ken Barker, Parul Awasthy, Jian Ni, and Radu Florian. 2021. Ibm mnlp ie at case 2021 task 2: Nli reranking for zero-shot text classification. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 193–202.

Hilda Bastian, Paul Glasziou, and Iain Chalmers. 2010. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9):e1000326.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2020. Semantic re-tuning with contrastive tension. In *International conference on learning representations*.

MohammadReza Davari. 2020. *Neural Network Approaches to Medical Toponym Recognition*. Ph.D. thesis, Concordia University.

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. 2022a. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16712–16721.

MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. 2022b. Reliability of cka as a similarity measure in deep learning. *arXiv preprint arXiv:2210.16156*.

MohammadReza Davari, Leila Kosseim, and Tien Bui. 2020. Timbert: toponym identifier for the medical domain based on bert. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 662–668.

MohammadReza Davari, Leila Kosseim, and Tien D Bui. 2019. Toponym identification in epidemiology articles–a deep learning approach. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 26–37. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Eric Lehman, Ben Nye, Iain J Marshall, and Byron C Wallace. 2020. Evidence inference 2.0: More data, better models. *arXiv preprint arXiv:2005.04177*.

Farhood Farahnak, Elham Mohammadi, MohammadReza Davari, and Leila Kosseim. 2021. Semantic similarity matching using contextualized representations. In *Canadian Conference on AI*, volume 1.

Farhood Farahnak, Laya Rafiee, Leila Kosseim, and Thomas Fevens. 2020. Surface realization using pretrained language models. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 57–63.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Donal Landers, and André Freitas. 2023a. Nli4ct: Multi-evidence natural language inference for clinical trial reports. *arXiv preprint arXiv:2305.03598*.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. 2024. Is cosine-similarity of embeddings really about similarity? *arXiv preprint arXiv:2403.05440*.

Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. 2020. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1):17.

Zachary Yang, Yasmine Maricar, Mohammadreza Davari, Nicolas Grenon-Godbout, and Reihaneh Rabbany. 2023. Toxbuster: In-game chat toxicity buster with bert. *arXiv preprint arXiv:2305.12542*.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. 2019. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*.

MALTO at SemEval-2024 Task 6: Leveraging Synthetic Data for LLM Hallucination Detection

Federico Borra* **Claudio Savelli*** **Giacomo Rosso**

Alkis Koudounas **Flavio Giobergia**

Politecnico di Torino, Italy

Abstract

In Natural Language Generation (NLG), contemporary Large Language Models (LLMs) face several challenges, such as generating fluent yet inaccurate outputs and relying on fluency-centric metrics. This often leads to neural networks exhibiting "hallucinations". The SHROOM challenge focuses on automatically identifying these hallucinations in the generated text. To tackle these issues, we introduce two key components, a data augmentation pipeline incorporating LLM-assisted pseudo-labelling and sentence rephrasing, and a voting ensemble from three models pre-trained on Natural Language Inference (NLI) tasks and fine-tuned on diverse datasets.

1 Introduction

Natural Language Generation (NLG) models are AI systems that use neural networks to produce human-like text. They have shown significant advancements in recent years, particularly with the advent of transformer-based architectures such as GPT (Generative Pre-trained Transformer) (Radford et al., 2018). These models offered unprecedented levels of fluency and coherence in generated text (Han et al., 2021). However, a critical challenge arises: these models can produce linguistically fluent but semantically inaccurate outputs, a phenomenon referred to as *hallucination* (Ji et al., 2023). This may also lead to the generation of offensive, misleading, or factually incorrect content, as highlighted in previous studies (Engstrom and Gelbach, 2020; Bender et al., 2021). Such issues could have profound repercussions, especially for marginalized or under-resourced communities (Surden, 2020; Volokh, 2023; Koudounas et al., 2024).

To address this challenge, the Shared-task on Hallucinations and Related Observable Overgeneration Mistakes (SHROOM) has been proposed at SemEval 2024 (Mickus et al., 2024). In particular, the Shared task aims to address the existing gap in assessing the semantic correctness and meaningfulness of NLG models. The ever-increasing adoption of such models makes it necessary to automatically detect and mitigate semantic hallucinations (Huang et al., 2023).

Some examples to tackle hallucination detection tasks in literature (Ji et al., 2023) are: (i) *Information Extraction and Comparison* between a generated text and a ground truth, (ii) *Natural Language Inference Metrics* that express the entailment between generated text and a ground truth or (iii) *Faithfulness Classification Metrics* that leverage upon knowledge-grounded datasets.

In this work, we address the SHROOM shared task by introducing an automatic pipeline of hallucination detection through the comparison between a generated text and a ground truth text. We propose enriching the original data available using different augmentation techniques, including LLM-aided pseudo-labeling and sentence rephrasing. Additionally, we suggest using an ensemble of three different approaches, incorporating a simple BERT-based classifier, a model trained through Conditioned Reinforcement Learning Fine Tuning (C-RLFT) (Wang et al., 2023), and a sequential model based on iterative fine-tuning. We show how this ensemble benefits from using different, complementary approaches, particularly in recall. Our methodology obtained an accuracy of 80.07% in the SemEval-Task 6 SHROOM.

1.1 Dataset

The dataset available for the SHROOM challenge is a collection of objects. Each object represents a solution of a generative language model to either of the three tasks. The first is *Definition Modeling* (DM) (Noraset et al., 2017), the task of providing a definition for a given word. The second is *Machine Translation* (MT), i.e., translating from a source

*These authors contributed equally to this work.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1678–1684

language to a target one: this has been shown to be a challenging task that can be addressed from both statistical (Koehn, 2009) and, in recent years, neural perspectives (Bahdanau et al., 2014; Giobergia et al., 2020). Finally, the task of *Paraphrase Generation* (PG) consists of paraphrasing, i.e., producing an alternative version, of a source sentence (Zhou and Bhat, 2021). Each solution has been annotated, based on its contents, as either a *hallucination* of the generative model or *not hallucination* by 5 human annotators.

For each object, the available information includes (i) the *source (src)*, which is the input text given to the generative language model, (ii) the *hypothesis (hyp)*, which represents the generated textual output of the model, and (iii) the *target (tgt)* which is the intended reference or "gold" text that the model is supposed to generate. Additionally, the task field indicates the type of task being solved, either DM, MT, or PG. The label, either *"hallucination"* or *"not hallucination"*, is determined through majority voting among five annotators, with *p(hal)* indicating the proportion of annotators who labeled the data point as a hallucination.

The gold (and augmented) data cardinalities are defined in Table 1. The training dataset comprises 500 instances with gold labels, denoted as $\mathcal{D}_g$, and 30,000 unlabelled instances, referred to as $\mathcal{D}_u$ (10,000 for each of the three tasks). The evaluation split contains 1,500 labelled samples, with 500 instances used for validation ($\mathcal{D}_v$) and 1,000 for testing ($\mathcal{D}_t$). We use the validation set for fine-tuning the ensemble layer (refer to Section 2.3), while the final test set provides overall results (see Section 3).

We further rephrase the original 500 labelled sentences of the training set ($\mathcal{D}_r$ in the table, see Section 2.1.2), while applying weak labelling to the 30,000 unlabelled instances ($\mathcal{D}_{pl}$, see Section 2.1.1).

2 Methodology

The main goal of this work is to propose a binary classification model to predict whether the answer to a given query is a hallucination or not. Figure 1 presents the main architecture adopted to address this task[1]. We propose (i) using a data augmentation pipeline (see Section 2.1) consisting of Large Language Model (LLM)-aided pseudo-labelling and sentence rephrasing and (ii) adopting an ensemble model (see Section 2.3) based on the results of three models, defined as follows:

- *Baseline* model, a binary classifier based on a semantic-aware embedding (e.g. BERT-based (Devlin et al., 2019)). The baseline model is presented in Section 2.2.1
- *C-RLFT* (Conditioned Reinforcement Learning Fine Tuning (Wang et al., 2023)), based on the introduction of pseudo-labels and augmented data, with different weighting schemes based on the quality of each data point. We cover C-RLFT in more detail in Section 2.2.2
- *Sequential* model, based on the iterative fine-tuning of the baseline model with increasingly higher-quality data, as detailed in Section 2.2.3

2.1 Data Augmentation

Due to the scarcity of data, we developed an approach to extend the number of labelled samples we could use to train our models. We specifically leverage two distinct techniques: pseudo-labelling and sentence rephrasing. Both approaches are based on LLMs and, as such, may themselves be subject to hallucinations or inaccuracies. As detailed next, we mitigate this problem by (1) using the C-RLFT technique (Wang et al., 2023), which involves assigning different weights to mixed-quality samples, and (2) with a sequential training that introduces different-quality labels at different training stages.

2.1.1 Pseudo Labeling

As stated in Section 1.1, only a small fraction of the dataset available is labelled. We introduce additional pseudo labels, as obtained by querying an LLM in a few-shot learning setting. Based on the hardware available, we tested several LLM models to assess the reliability of the pseudo labels produced (in terms of accuracy). We identified SOLAR (Kim et al., 2023) as being the best-performing model among the pool of candidates. Thus, we leverage it to generate synthetic labels for unlabelled data through a few-shot learning approach. We refer to this augmented dataset as $\mathcal{D}_{pl}$.

2.1.2 Sentence Rephrasing

We utilized sentence rephrasing based on GPT-4 as an additional data augmentation technique. We

[1]The code to replicate the experiments can be found at https://github.com/MAL-TO/shroom

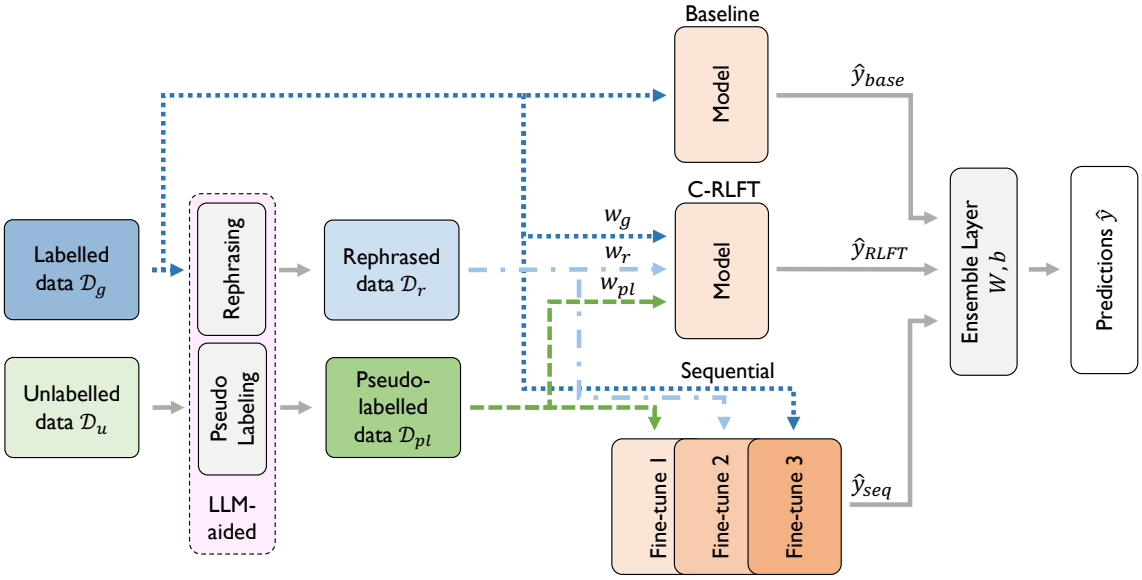


Figure 1: Pipeline architecture depicting data augmentation techniques and weighted ensemble of strategies.

do so by rephrasing both the model output and the target output of each gold sample. This approach aims to provide the model with diverse data while maintaining the reliability of the labels. We refer to this dataset as $\mathcal{D}_r$.

2.2 Models

We adopt an ensemble of three models, as described below. All models are based on DeBERTa (**?**). More specifically, we use a baseline model that has been fine-tuned in different ways.

2.2.1 Baseline

We employed a baseline model utilizing the DeBERTa encoder pre-trained on the Natural Language Inference (NLI) task, with a binary classification head. We fine-tune this model on the provided classification task using only data with the gold labels available, referred to as $\mathcal{D}_g$. The training approach involved minimizing the Binary Cross Entropy (BCE) loss.

We use the probability $p(hal)$ as the ground truth instead of the binary label. This is done to better reflect the distribution of votes of the human annotators in the output logits of the model.

2.2.2 C-RLFT

Conditioned Reinforcement Learning Fine Tuning (C-RLFT) is a technique that refines models using coarse-grained reward labels, allowing fine-tuning with both expert and sub-optimal data lacking preference labels. In our specific scenario, we fine-tuned the model by assigning different weights to data based on their label type, i.e., synthetic or gold. The weight assigned to each data sample influences the contribution to the final BCE loss.

We define a weighting scheme for the gold dataset $\mathcal{D}_g$, the pseudo-labelled dataset $\mathcal{D}_{pl}$ and the rephrased dataset $\mathcal{D}_r$, as follows:

$$w(x_i) = \begin{cases} w_g & \text{if } x_i \in \mathcal{D}_g \\ w_r & \text{if } x_i \in \mathcal{D}_r \\ w_{pl} & \text{if } x_i \in \mathcal{D}_{pl} \end{cases}$$

We choose weights $w_g > w_r > w_{pl}$. In this way, we aim to assign a higher importance to gold labels due to their reliability. The lowest weight is assigned to the pseudo-labelled points because of the lower quality of the automatically assigned labels. An intermediate weight is given to rephrased sentences due to the higher quality of the ground truth w.r.t. the pseudo-labelled points. The weighted loss is thus defined as follows, for a point x_i with ground truth y_i, as computed for a binary classifier $f(\cdot)$:

$$\begin{aligned} wBCE(x_i, y_i) = &- w(x_i) \cdot (y_i \; log \; f(x_i) \\ &+ (1 - y_i) \; \log(1 - f(x_i))) \end{aligned}$$

2.2.3 Sequential

The third model used is based on a sequential strategy that uses both generated and augmented data. We introduce three fine-tuning steps, performed sequentially on the initial model. The model initially underwent fine-tuning using the pseudo-labelled dataset $\mathcal{D}_{pl}$, which is the lowest-quality dataset

among the three available. Subsequently, we fine-tuned the resulting model on the rephrased data $\mathcal{D}_r$, which benefits from the original, correct, labels. The final fine-tuning step is then executed on the golden truth dataset $\mathcal{D}_g$. This approach is inspired by curriculum learning (Soviany et al., 2022), with data being ordered by veracity instead of difficulty.

This strategy aims to enhance the model's understanding of the task by starting with a substantial amount of data, including the less reliable synthetic labels, and progressively updating the model parameters with increasingly consistent data. This sequential approach allows the model to first adapt to the task using a broader dataset and then refine its knowledge with the highest quality data available.

2.3 Ensemble

The final step in the proposed pipeline involves creating an ensemble of results from the previously-introduced techniques, which has already proven to be effective in other NLP tasks (Jia et al., 2023; Koudounas et al., 2023). We trained three distinct models (*baseline*, *C-RLFT*, *sequential*) with specific strategies, and we generated their outputs ($\hat{y}_{base}$, $\hat{y}_{RLFT}$, $\hat{y}_{seq}$) on a validation set of previously unseen gold data. We obtain a single result $\hat{y}$ from the previous ones by using a single-layer network ($W \in \mathbb{R}^3$ and $b \in \mathbb{R}$), as follows:

$$\begin{aligned} \hat{y}_{models} &= (\hat{y}_{base}, \hat{y}_{RLFT}, \hat{y}_{seq}) \\ \hat{y} &= \sigma(W^\intercal \hat{y}_{models} + b) \end{aligned} \tag{1}$$

This network is trained to predict a single output from the three models' predicted probabilities. We trained this network by minimizing a BCE function.

3 Experimental Results

This section presents the experimental setup used, and the main results obtained.

3.1 Experimental Setup

The dataset used to train and validate the model is the one made available in the SHROOM challenge's model agnostic track (refer to Section 1.1). The augmentations are specified in Section 2.1.

For the model backbone and synthetic labelling we leverage Huggingface pre-trained models[2]. We also leverage *GPT-4* for sentence rephrasing. All the experiments' results are obtained based on 5 different runs. For C-RLFT, we identified the best performance for weights $w_g = 1.01, w_r = 0.4, w_{pl} = 0.1$.

Dataset Type	Label	Split	#Samples
$\mathcal{D}_g$	yes	Train	500
$\mathcal{D}_r$	yes	Train	500
$\mathcal{D}_u$	no	Train	30,000
$\mathcal{D}_{pl}$	weak	Train	30,000
$\mathcal{D}_v$	yes	Val	500
$\mathcal{D}_t$	yes	Test	1000

Table 1: Dataset type, labelling, and number of instances for each considered split.

3.2 Model performance

We summarize the results obtained on the test set in Table 2. We report the results in terms of F_1 score, precision, and recall on the "Hallucination" class, as well as overall accuracy.

We use as backbone both DeBERTa and a version of DeBERTa that has been fine-tuned on the Machine Natural Language Inference (MNLI) task. Further discussions on the choice of the backbone are presented in Section 3.3.

Section 3.4 highlights the result differences for each of the considered strategies and includes additional considerations on the ensemble of the approaches. Finally, we provide qualitative examples of the results in Section 3.5.

3.3 Backbone impact

We start by examining the differences between two backbone models, both fine-tuned on the gold data only – these are referred to as the *baseline* models in Table 2.

There is a notable increase of 0.09 in the F_1 score for the MNLI-fine-tuned model compared to the original DeBERTa. Interestingly, all the proposed DeBERTa-based approaches are still outperformed by the baseline DeBERTa+MNLI-based model (although to a lesser extent). This highlights the close relationship between the tasks of *Hallucination Detection* and *Natural Language Inference*.

3.4 Strategies comparisons

The *baseline* strategy, which utilizes all available labelled gold data, establishes a lower bound in the expected performance. Both *C-RLFT* and *sequential training* exhibit substantial performance

[2]We use *deberta-xlarge* and *deberta-xlarge-mnli* as encoders, *TheBloke/SOLAR-10.7B-Instruct-v1.0-GPTQ* for pseudo labelling.

Model	Method	F_1 score	Precision	Recall	Accuracy
DeBERTa	Baseline	0.6207±0.0808	0.7112±0.0661	0.5588±0.1562	0.7254±0.0231
DeBERTa	C-RLFT	0.6182±0.0857	0.8081±0.0939	0.5089±0.1574	0.7476±0.0245
DeBERTa	Sequential	0.7075±0.0394	**0.8169±0.0396**	0.6253±0.0690	**0.7898±0.0194**
DeBERTa	Ensemble	**0.7119±0.0272**	0.7918±0.0402	**0.6474±0.0466**	0.7867±0.0171
D.+MNLI	Baseline	0.7138±0.0253	0.7420±0.0319	**0.6882±0.0372**	0.7753±0.0178
D.+MNLI	C-RLFT	0.6146±0.0917	**0.8410±0.0706**	0.4900±0.1376	0.7528±0.0302
D.+MNLI	Sequential	0.7320±0.0229	0.8177±0.0233	0.6628±0.0329	**0.8024±0.0141**
D.+MNLI	Ensemble	**0.7371±0.0223**	0.8016±0.0347	0.6829±0.0425	0.8017±0.0143

Table 2: Performance metrics for DeBERTa and DeBERTa + MNLI models. Best scores are highlighted in bold, and second-best are underlined.

src	hyp	tgt	Target $\mathbf{p(hal)}$	$\hat{\mathbf{p}}\mathbf{(hal)}$
Король Харальд Гормссон, более известный как Харальд Синезубый, ввёл в Дании христианство.	King Harald Hormsson, better known as Harald Sinezubii, introduced Christianity to Denmark.	King Harald Gormsson, better known as “Harald Bluetooth", introduced Christianity to Denmark.	0.40	0.40
Why’d you got to go and do that?	Why did you have to go do that?	Why would you say that?	0.00	0.91

Table 3: Examples of correctly and wrongly predicted as *“Hallucination”* or *“Not Hallucination”*. The model output is *p(hal)* and must be confronted with gold *p(hal)*. The first example proposed is a Russian to English *Machine Translation* (MT), and the second is an English *Paraphrase Generation* (PG).

improvements.

Regarding the *ensemble* strategy, the results in terms of F_1 score outperform individual techniques. We observe a trade-off where the precision of the final result is slightly compromised in exchange for an improved recall. This suggests that the *ensemble* effectively identifies instances of hallucination overlooked by the standalone approaches. These advantages are consistent across both backbones implementations, with and without the additional MNLI fine-tuning.

In a setting where detected hallucinations are shown to the final user with a warning, we argue that the recall is a metric of greater interest (w.r.t. precision). A false negative could be potentially harmful since final users are not warned of the presence of possible hallucinations. A false positive would raise a warning that may be inspected by the final user and safely ignored.

The weights learned for the ensemble layer, based on Equation 1, are $W = (0.52, 1.7, 1.82)$, $b = -1.7$. This shows how both C-RLFT and the sequential models are weighted similarly and more heavily w.r.t. the baseline. The baseline is assigned a non-zero weight: it is considered, although to a lesser extent, in the final vote. The negative bias implies a learned prior: without further knowledge, the initial prediction is of a negative one (i.e., the majority class).

3.5 Qualitative Example

Table 3 demonstrates the effectiveness of the applied strategies through some qualitative examples. We specifically showcase the sentences with the minimum (first row) and maximum (second row) errors.

The first instance depicts a partial hallucination, attributed to the transliteration of “Sinezubii” instead of the translation “Bluetooth,” which is absent from the translation hypothesis. In the second example, despite a paraphrased similarity between the source and hypothesis, the target introduces an action (“saying”) not present in the source (“doing”). As such, we argue that this might be a case of incorrectly labelled ground truth.

4 Conclusions

This work tackles the SHROOM Task 6 challenge at SemEval 2024, focusing on semantic hallucination in NLG models. We propose an automatic

pipeline for hallucination detection, utilizing data augmentation and an ensemble of three different methodologies. The ensemble of the approaches obtained an accuracy of 80.07% in the task's leaderboard. Particular attention should also be paid to the results obtained with the novelty method *sequential*, which was able to outperform the results of the other two methods due to the proposed sequential training.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

David Freeman Engstrom and Jonah B Gelbach. 2020. Legal tech, civil procedure, and the future of adversarialism. *U. Pa. L. Rev.*, 169:1001.

Flavio Giobergia, Luca Cagliero, Paolo Garza, Elena Baralis, et al. 2020. Cross-lingual propagation of sentiment information based on bilingual vector space alignment. In *EDBT/ICDT Workshops*, pages 8–10.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Jianguo Jia, Wen Liang, and Youzhi Liang. 2023. A review of hybrid and ensemble in deep learning for natural language processing. *arXiv preprint arXiv:2312.05589*.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2023. Solar 10.7b: Scaling large language models with simple yet effective depth up-scaling.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Alkis Koudounas, Flavio Giobergia, Irene Benedetto, Simone Monaco, Luca Cagliero, Daniele Apiletti, Elena Baralis, et al. 2023. baρtti at geolingit: Beyond boundaries, enhancing geolocation prediction and dialect classification on social media in italy. In *CEUR Workshop Proceedings*. CEUR.

Alkis Koudounas, Eliana Pastor, Giuseppe Attanasio, Vittorio Mazzia, Manuel Giollo, Thomas Gueudre, Elisa Reale, Luca Cagliero, Sandro Cumani, Luca de Alfaro, Elena Baralis, and Daniele Amberti. 2024. Towards comprehensive subgroup performance analysis in speech models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey.

Harry Surden. 2020. The ethics of artificial intelligence in law: Basic questions. *Forthcoming chapter in Oxford Handbook of Ethics of AI*, pages 19–29.

Eugene Volokh. 2023. Chatgpt coming to court, by way of self-represented litigants. *The Volokh Conspiracy*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 5075–5086.

Maha Bhaashya at SemEval-2024 Task 6: Zero-Shot Multi-task Hallucination Detection

Patanjali Bhamidipati[†]
IIIT Hyderabad
patanjali.b@research.iiit.ac.in

Advaith Malladi[†]
IIIT Hyderabad
advaith.malladi@research.iiit.ac.in

Manish Shrivastava
IIIT Hyderabad
m.shrivastava@iiit.ac.in

Radhika Mamidi
IIIT Hyderabad
radhika.mamidi@iiit.ac.in

Abstract

In recent studies, the extensive utilization of large language models has underscored the importance of robust evaluation methodologies for assessing text generation quality and relevance to specific tasks. This has revealed a prevalent issue known as hallucination, an emergent condition in the model where generated text lacks faithfulness to the source and deviates from the evaluation criteria. In this study, we formally define hallucination and propose a framework for its quantitative detection in a zero-shot setting, leveraging our definition and the assumption that model outputs entail task and sample specific inputs. In detecting hallucinations, our solution achieves an accuracy of 0.78 in a model-aware setting and 0.61 in a model-agnostic setting. Notably, our solution maintains computational efficiency, requiring far less computational resources than other SOTA approaches, aligning with the trend towards lightweight and compressed models.

1 Introduction

The contemporary landscape of Natural Language Generation (NLG) is marked by a confluence of complexities, wherein two primary challenges emerge as focal points of concern. Firstly, the prevalent neural models within NLG frameworks consistently produce outputs that exhibit linguistic fluency yet suffer from inaccuracies (Huang et al., 2023). Secondly, the current evaluation metrics, vital for evaluating the effectiveness of NLG systems, demonstrate a significant inclination towards fluency measures while neglecting to prioritize accuracy. So, this highlights the need to consider the "truthfulness" of the model's output, i.e its alignment with the source to ensure a comprehensive assessment.(Dale et al., 2022)

In the realm of NLG applications, the criticality of output accuracy cannot be overstated. A divergence between the fluency and factual correctness of generated content not only undermines the utility of NLG systems but also engenders substantial risks across various domains. Consider, for instance, the domain of machine translation,the production of seemingly plausible yet inaccurate translations not only compromises the integrity of the translated content but also defeats the purpose of facilitating correct translations.

Likewise, in tasks like definition modeling and paraphrase generation, where accurately conveying semantic meaning is crucial, the presence of incorrect outputs presents notable challenges in upholding the integrity and dependability of the generated content.

2 SHROOM Dataset

SHROOM (a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes) dataset is a task-based hallucination detection dataset which is divided into two major categories:

2.1 Model Aware (MAw)

Model Aware (MAw) refers to situations where the model under study is known.

2.2 Model Agnostic (MAg)

Model Agnostic (MAg) refers to situations where the model under study is not known.

The dataset encompasses three major Natural Language Generation tasks, namely:

1. Definition Modeling (DM): In this task, models are trained to generate a definition for a given example in context.

[†]The authors contributed equally to this work.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1685–1689

2. Machine Translation (MT): In this task, models aim to generate translations of the given samples.

3. Paraphrase Generation (PG): In this task, models aim to produce paraphrases of the given text samples.

Further, each sample in the train set is populated with information such as task ($task$): indicating what objective the model was trained for, source (src): the input passed to the models for the generation, target (tgt): the intended reference "gold" standard text that the model ought to generate, hypothesis (hyp): the actual model production,also the model-aware dataset is populated with model name ($model$) used for the task, with the val set additionally being populated with majority-based gold-label ($label$), based on the annotator labels along with the probability values of the sample being hallucination ($p(Hallucination)$) based on the proportion of annotators who claim that the sample is an hallucination.

3 Definitions

As described earlier, the SHROOM shared task encompasses of three different taks, Definition Modelling (DM), Paraphrase Generation (PG), and Machine Translation. We define the **Hallucination** in the context of the specific task at hand. Defining hallucinations individually in the context of a specific task enables detecting hallucinations quantitatively and qualitatively. We offer distinct definitions of hallucinations and methodologies for detecting hallucinations within the context of each of the aforementioned task.

In the context of definition modelling, the model is expected to generate the definition of a word which has been used in the provided context by making use of distributional semantics. Definition modelling models such as flan-t5-definition-en-base (Giulianelli et al., 2023) are not fully capable of making use of distributional semantics to define a word as used in a context. In a sample where the word W has been used in a setting contrasting to the definition the model has learnt during its training process, the models fails to provide a contextual definition of the word W. Examples for the same have been demonstrated in Table 1 and Table 2. We observe the model outputs a definition of word W which is very similar what it has learnt during its training process. Based on this observation, we assume that the targets provided in the SHROOM dataset have been extracted from the training data of the definition modelling dataset. With this assumption, we define **"Hallucination to be an instance where the output generated by the definition modelling model *does not* entail the target output."** Thereby reducing the hallucination detection task to a Natural Language Inference task in the context of definition modelling.

In the context of paraphrase generation and machine translation, the model's inputs and outputs are anticipated to exhibit semantic equivalence. If the generated paraphrase or translation diverges from semantic equivalence with the source text, they are deemed imperfect paraphrases or translations. Therefore, in the context of paraphrase generation and machine translation, we define **"Hallucination to be an instance where the paraphrase or translation generated by the model is *not semantically equivalent* to the source."** This reduces the hallucination detection in the context of given tasks to a semantic equivalence detection task, which could also be framed as bidirectional entailment detection, a variation of the Natural Language Inference task.

The aforementioned definitions of hallucination allow us to simplify the hallucination detection task to a Natural Language Inference task, thereby enabling us to qualitatively and quantitatively detect hallucinations.

In a more generic setting, we provide a definition of hallucinations that can be adapted to any task to effectively detect them. We define **"hallucinations as instances where the output generated by the model is *not faithful* to the input or the training data of the model. If the model generates information that is contradictory to the model's training data or the input to the model, it can be termed as a hallucination."**

4 Methodology

Grounding to the above definitions, the experimental setup we designed goes on to quantify the alignment of the model's output (hyp) with either the source (src) or the target (tgt) based on the task ($task$) the data sample corresponds to.

We propose that examining the **entailment** relationship between the model's output (hyp) and either the source (src) or the target (tgt) (which is

Example 1: Definition Modeling (DM)
Model Input: I went into the *water bottle* to withdraw cash. What is the definition of *water bottle*?
Model Output: A container for holding liquids.
Expected Output: A financial institution such as a bank or ATM to withdraw cash
Model: flan-t5-definition-en-base

Table 1: A Table showing distinction between the model output and the expected output where the model fails to understand the contextual definition of the term *water bottle*

Example 2: Definition Modeling (DM)
I jumped into the flaxcron to do some swimming. What is the meaning of *flaxcron*?
Model Output: A slender, slender
Expected Output: A pool of water.
Model: flan-t5-definition-en-base

Table 2: A Table showing distinction between the model output and the expected output where the model fails to understand the contextual definition of the term *flaxcron*

also inherently linked to the source (src)), depending on the task, sheds light on data samples that are **not** "detached" from the source. Consistent with our initial hypothesis that hallucinations occur when samples are "detached" from the source, this approach based on Natural Language Inference (NLI) can effectively aid in hallucination classification.

- In the context of definition modelling, **if the** hyp **does not entail the** tgt, the sample has been classified as **Hallucination**.
- In the context of machine translation and paraphrase generation, we check equivalence through bidirectional entailment. **If the** hyp **does not bidirectionally entail the** src, the sample has been classified as **Hallucination**
- In the context of machine translation and paraphrase generation, we verify our hypothesis of semantic equivalence between the src and hyp by comparing the performance metrics in the case of both unidirectional and bidirectional entailment.

Recent research heavily relies on large language models (LLMs) to benchmark various natural language understanding and generation (NLG) tasks. However, this practice extends to hallucination detection as well, which we find ironic and counterproductive, considering LLMs' inherent tendency to hallucinate. Using LLMs for hallucination detection presents two major drawbacks. Firstly, their computational demands are significant, making them an expensive solution (Bai et al., 2024). Secondly, the lack of complete interpretability in LLMs renders them unreliable for this task (Singh et al., 2024).

5 Results

After several experiments with the above methodology, leveraging the accuracy and the Spearman correlation (ρ) metrics, we have bench-marked the hallucination detection task on the SHROOM validation and test sets to achieve an accuracy of 0.78 in model-aware and 0.61 on model-agnostic test sets respectively. For our analysis let us take only the accuracy metric into account.
The bench-marking saw a utilisation of open-source pre-trained Natural Language Inference (NLI) models available on Hugging Face. Several experiments brought out interesting observations which are worthy discussing.

We evaluated the following models:

1. MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli **(DeBERTa-1)** (He et al., 2020)
2. MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7 **(DeBERTa-2)** (He et al., 2020)

Model	Unidirectional	Bidirectional
DeBERTa - 1	0.783567	0.755511
DeBERTa - 2	0.765531	0.717435
BART - 1	0.769539	0.733467
RoBERTa - 1	0.757515	0.727455

Table 3: Model-agnostic evaluation on (Uni vs Bi) directional entailment.

Model	Unidirectional	Bidirectional
DeBERTa - 1	0.596806	0.570859
DeBERTa - 2	0.576846	0.586826
BART - 1	0.610778	0.568862
RoBERTa - 1	0.612774	0.584830

Table 4: Model-aware evaluation on (Uni vs Bi) directional entailment.

3. ynie/bart-large-snli_mnli_fever_anli_R1_R2_R3-nli **(BART-1)** (Lewis et al., 2019)

4. ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli **(RoBERTa-1)** (Nie et al., 2020)

5.1 Definition Modelling

For the task of definition modelling, our approach achieves a peak accuracy of 0.748663 using the DeBERTa-2 model in a model-agnostic setting and a peak accuracy of 0.755319 using the RoBERTa-1 model in a model-aware setting. These results are in accordance with our hypothesis that when the model hallucinates, it does not entail the target.

5.2 Paraphrase Generation and Machine Translation

For the paraphrase generation and machine translation tasks, the observed results confirm our hypothesis that if the source (src) and hypothesis (hyp) are not semantically equivalent, the hypothesis is a hallucination. In hallucination detection for the paraphrase generation task, we observe that bidirectional entailment (semantic equivalence) outperforms the unidirectional entailment approach for all models. Similar results can also be observed for the machine translation task. This provides evidence that in machine translation and paraphrase generation, hallucinations can be detected by checking for semantic equivalency between the source and hypothesis.

Model	Unidirectional	Bidirectional
DeBERTa - 1	0.728	0.752
DeBERTa - 2	0.624	0.68
BART - 1	0.696	0.72
RoBERTa - 1	0.712	0.752

Table 5: Accuracy validation on **PG** task

Model	Unidirectional	Bidirectional
DeBERTa - 1	-	-
DeBERTa - 2	0.722	0.754
BART - 1	-	-
RoBERTa - 1	-	-

Table 6: Accuracy validation on **MT** task.

Model	DM	MT	PG
DeBERTa - 1	0.721925	0.855615	0.768
DeBERTa - 2	0.748663	0.823529	0.704
BART - 1	0.748663	0.844920	0.688
RoBERTa - 1	0.711230	0.834224	0.712

Table 7: Model-agnostic evaluation on individual tasks.

5.3 Overall Analysis

In the model-agnostic setting, we achieve a peak accuracy of 0.783567 using the DeBERTa-1 model and a peak accuracy of 0.612774 in a model-aware setting using the RoBERTa-1 model. These scores exhibit satisfactory performance of models pretrained on the Natural Language Inference task for Hallucination Detection.

6 Conclusion

Our work makes two significant contributions to the study of hallucinations in language models. First, we provide a concrete definition of the term "hallucination," enabling both qualitative and quantitative study and detection of such phenomena. Second, we offer a computationally efficient approach to detect hallucinations in tasks such as definition modeling, machine translation, and paraphrase generation. We frame the hallucination detection task as a function of the input to the generation model and the data used to train it. Our definitions and approaches also provide a framework that can be utilized for hallucination detection in

various Natural Language Generation tasks across the spectrum.

References

Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, Carl Yang, Yue Cheng, and Liang Zhao. 2024. Beyond efficiency: A systematic survey of resource-efficient large language models.

David Dale, Elena Voita, Loïc Barrault, and Marta R. Costa-jussà. 2022. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better.

Mario Giulianelli, Iris Luden, Raquel Fernandez, and Andrey Kutuzov. 2023. Interpretable word sense representations via definition generation: The case of semantic change analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3130–3148, Toronto, Canada. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. 2024. Rethinking interpretability in the era of large language models.

Team art-nat-HHU at SemEval-2024 Task 8: Stylistically Informed Fusion Model for MGT-Detection

Vittorio Ciccarelli[†]
Heinrich-Heine-Universität Düsseldorf (HHU)
vittorio.ciccarelli@hhu.de

Cornelia Genz[†]
Heinrich-Heine-Universität Düsseldorf (HHU)
cornelia.genz@hhu.de

Nele Mastracchio[†]
Heinrich-Heine-Universität Düsseldorf (HHU)
nele.mastracchio@hhu.de

Wiebke Petersen[†]
Heinrich-Heine-Universität Düsseldorf (HHU)
wiebke.petersen@hhu.de

Anna Sophia Stein[†]
Heinrich-Heine-Universität Düsseldorf (HHU)
anna.stein@hhu.de

Hanxin Xia[†]
Heinrich-Heine-Universität Düsseldorf (HHU)
hanxin.xia@hhu.de

Abstract

This paper presents our solution for subtask A of shared task 8 of SemEval 2024 for classifying human- and machine-written texts in English across multiple domains. We propose a fusion model consisting of a RoBERTa-based pre-classifier and two MLPs that have been trained to correct the pre-classifier using linguistic features. Our model achieved an accuracy of 85%.

1 Introduction

After rapid developments in large language models (LLMs) and generative AIs in the last years, the detection of machine-generated content has become one focus of study as deepfakes, machine-generated lawyer statements and even libel suits (Superior Court of Gwinnett County) concerning language machines stress the importance of detecting texts not written by humans. The SemEval shared task 8 in 2024 aims at multi- and monolingual machine-generated text (MGT) detection from various domains by multiple models.

For the monolingual English data in subtask A (Wang et al., 2024) we propose a fusion model built using pre-trained RoBERTa word embeddings specialized for AI-generated text detection and correction MLP classifiers, supported by the additional computation of linguistic, stylistic and probabilistic features selected based on their informational value. With this system design, our model ranked at position 25 out of 124 with an 0.855 accuracy score on the task. The only data used for training was the one provided by the organizers without further data augmentation. Because of the different distributions of the data in the development and test data sets several strategies were tested and a fusion model was chosen as the best strategy. The fine-tuned RoBERTa Base OpenAI Detector alone performed well but developed a bias towards the machine class. To stabilize the model linguistic, probabilistic and stylistic features were added, which improved the overall F1 score of the fusion architecture.

2 Background

Over the last years, numerous approaches have been proposed to tackle the task of MGT detection. Some models, such as DetectMGT (Mitchell et al., 2023), focus on detecting texts from a specific source, such as GPT-family LLMs, while other approaches are specialized in texts from a specific genre, such as Shijaku and Canhasi (2023) for TOEFL essays. Other architectures, like ensemble models combining different classifiers (del Campo-Ávila et al., 2007) have been successfully used for machine-generated text detection to improve out-of-distribution performance (Lai et al., 2024).

Guo et al. (2023) show that, overall, deep-learning approaches, and in particular a RoBERTa-based-detector, are one of the best individual models for MGT detection. The RoBERTa-based-detector was shown to be particularly robust against oov scenarios in both Chinese and English, compared to a machine learning model. Moreover, Wang et al. (2023) and He et al. (2024) conducted large-scale bench-marking on existing approaches for MGT detection across multiple domains, models, and languages and concluded that the RoBERTa language model, especially the variants that have been optimized for AI detection tasks, consistently outperforms most other methods across evaluation metrics.

[†]Equal contribution.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1690–1697

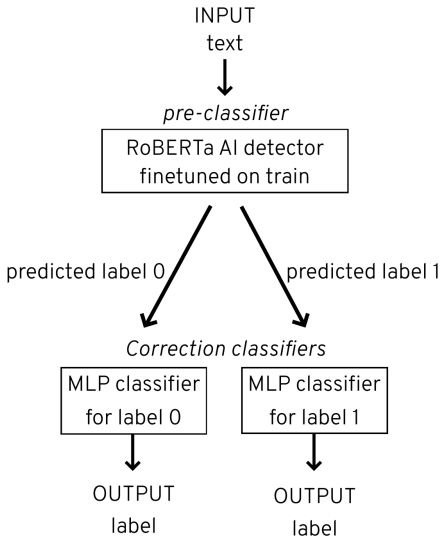


Figure 1: Model architecture used to obtain submission results.

3 System overview

Our system is based on a pre-classifier that is a RoBERTa model fine-tuned for AI generated text detection. In order to correct the predictions of the pre-classifier, two correction classifiers have been trained that are based on linguistic, stylistic and probabilistic features. An overview of the system setup is given in Figure 1.

Our RoBERTa pre-classifier is based on the RoBERTa Base OpenAI Detector[1] (Solaiman et al., 2019), a RoBERTa model fine-tuned for AI generated text detection. This model has been further fine-tuned on 10% of the training data. A prediction was then generated for each text in the train, dev and test set. In order to improve the predictions, two correction Multi-layer Perceptron (MLP) classifiers, one for each label, were trained on the training and development data of their respective label (see Figure 1) as well as on a range of features outlined in Section 4.2.1. To generate the final classification, all texts were classified again by the correction MLP that corresponded to the label predicted by the RoBERTa pre-classifier. This provided an opportunity for the more specialized classifier to correct the initial prediction.

4 Experimental setup

The M4 dataset consists of both machine (label 1) and human-generated texts (label 0). The dataset features texts from six different LLM generators (Davinci, chatGPT, Dolly, Cohere, BLOOMz and GPT4) and five different genres (Reddit, WikiHow, ArXive, Wikipedia, and peerRead). Participants were provided first with a train and dev set and later with a test set with 119,757, 5,000, and 34,272 texts in total, respectively.

Roughly 53% of the documents in the train set are machine generated (DaVinci: 14,343, chatGPT: 14,339, Dolly: 14,046, Cohere: 13,678), and 47% are human-written. In the dev set, exactly half of the texts were machine-generated by the BLOOMz model, the other half was human-written. The test set contains 18,000 (53%) machine-generated texts from Davinci, chatGPT, Dolly, Cohere, BLOOMz and GPT4 (3,000 texts each) and 16,272 (47%) human-written texts. An overview of the data is provided in Table 1. Since we did not include genre- or machine-specific information for our approach, this information is excluded from the table.

	train	dev	test
machine	53%	50%	53%
human	47%	50%	47%
total texts	119,757	5,000	34,272

Table 1: Label distribution across train, dev and test set.

4.1 RoBERTa pre-classifier

We used a fine-tuned RoBERTa Base OpenAI Detector as our pre-classifier. Because the OpenAI Detector had already been fine-tuned for human-machine classification, and to facilitate replication of the experiment, we used only 10% of the training data to further fine-tune the model[2].Training was done for 3 epochs with a learning rate of $2e^{-5}$ on Google Colab using a T4 run-time and took 45 minutes.

4.2 Correction classifiers

4.2.1 Feature extraction

To capture characteristics of machine-generated and human-written texts, the data was analyzed for various linguistic features. Altogether 70 features, widely used in NLP and easy to compute, were extracted, 35 of which exhibited a high to medium correlation with the gold label (see Table 4 in the Appendix). All features were computed on a 24GB

[1] https://huggingface.co/openai-community/roberta-base-openai-detector

[2] The data for fine-tuning consisted of 2,000 texts of each author category (Davinci, Cohere, Dolly, chatGPT, and human).

RAM machine with a Ryzen 7 7730U, which took up to 6 hours for all texts depending on the feature.

Count-based features. The texts were split into words and punctuation using regular expressions to derive the following features: mean sentence length, ratio of punctuation to words, ratio of word types to tokens, ratio of vowels to words and mean word length. The NLTK stopword list was used to get the ratio of content words to other words. Additionally the number of hapax legomena per text and the number of negation words (manually compiled list) per text were computed.

Syntactic features. All texts were POS-tagged with the NLTK part of speech tagger to compute syntactic features: ratio of nouns, verbs, adjectives, adpositions, adverbs, conjunctions, numerals, pronouns and determiners to words alltogether, ratio of adjectives to nouns and ratio of verbs to nouns.

Using the dependency parser of Spacy (Honnibal and Montani, 2017) we extracted the maximum depth of a dependency tree, mean depth of all dependency trees in a text, and number of passive constructions (determined by the number of *nsubjpass* POS tags) per sentence.

Frequency features. To capture whether the texts differ in word use, the logarithmic frequency of all content words in the human texts were computed. Additionally, lists of frequent words (frequency $\geq$ 12) and hapax legomena (frequency = 1) have been computed. From this the following features were extracted for all texts: mean log frequency of content words, ratio of frequent words to content words, ratio of hapax legomena to content words and number of hapax legomena.

Additionally, we used the Wiktionary frequency lists for English[3] and extracted a list of high frequency words (top 10%), mid-high frequency words (top 20%) as well as field specific word lists, namely the most frequent words in fantasy texts and in Wikipedia articles. For each list and each text in the datasets we extracted the ratio of words belonging to the lists to the content words as a feature.

Word difficulty features. The CEFR-J[4] project provides vocabulary lists for the different proficiency levels of the Common European Framework of Reference for Languages (CEFR)[5]. We used these lists[6] to compute the following features: ratio of A1/A2/B1/B2/C1/C2-level words to content words. This was done twice: once on the basis of the stemmed and once on the lemmatized words. We used the Porter stemmer and the WordNet lemmatizer from NLTK.

Stylistic and sentiment features. A number of features concerning text style and text sentiment were extracted. Using the same method as for the difficulty features above, we extracted the ratio of words in the list of negative opinion words compiled by Liu et al. (2005) as well as the readability score of the texts according to the Flesch reading-ease test[7] . The other features in this subset have been extracted by using available fine-tuned classifiers. Emotion English DistilRoBERTa-base[8] is a classifier that predicts Ekman's six basic emotions, plus a neutral class (cf. Hartmann, 2022). The logit for each class provides one feature (anger, disgust, fear, joy, neutral, sadness, surprise). As a standard sentiment analyzer we used the sentiment-analysis-pipeline from Hugging Face[9] and, using the logits, extracted two features (positive, negative). Analogously, the features 'formal' and 'informal' were extracted using the formality ranker by Babakov et al. (2023), which is a RoBERTa model trained to predict to which register a sentence belongs. Finally, we used a toxicity classification model[10] that is a RoBERTa model fine-tuned to predict whether a text is toxic or not.

Features extracted from the pre-classifier. In order to inform the correction classifiers on the basis of the decision of the pre-classifier, we extracted the logits and the last hidden state of our RoBERTa pre-classifier for each text. The last hidden states were reduced from 768 to 2 dimensions using PCA (principal component analysis) and UMAP (uniform manifold approximation and projection). For UMAP the hyperparameters `min_dist`, `n-neighbors` and `metric` were tuned by a combination of random search and grid search

[3] https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/English

[4] https://www.cefr-j.org/

[5] https://www.coe.int/en/web/common-european-framework-reference-languages

[6] https://github.com/openlanguageprofiles/olp-en-cefrj/tree/master

[7] https://github.com/textstat/textstat

[8] https://huggingface.co/j-hartmann/emotion-english-distilroberta-base

[9] https://huggingface.co/

[10] https://huggingface.co/s-nlp/roberta_toxicity_classifier

and evaluated on the accuracy of a logistic regression classifier that predicts the label from the 2 dimensions. The extracted features included in our feature set are the logits, the 2-dimensional PCA representation of the last hidden state and the 2 UMAP dimensions gained by setting `min_dist` to 0.01 and `n-neighbors` to 100. We kept two metrics, namely `cosine` and `jaccard`.

4.2.2 Feature selection

To account for the variability in features, we initially scaled all 70 features using the Standard Scaler from scikit-learn (Pedregosa et al., 2011). During the collection of the 70 features, no attention was paid to whether they contained quasi-duplications. Features which were highly correlated with other features (>0.9) were removed subsequently using a correlation matrix. After this removal, 51 features remained.

In the next step, only features with high or medium correlation with the gold label (Pearson correlation ≥ 0.1 or ≤ -0.1) were retained in order to choose the features most relevant for the classification task. Table 4 in the Appendix shows all features (including those which are highly correlated to each other) that have at least a medium positive or negative correlation with the gold label. After both selection steps, 26 features remained (see Table 5).

4.2.3 Model selection and training

In a comparison of various classifiers from scikit-learn (i.e. Random Forest, Logistic regression), MLPs performed best in most settings: whether trained on all features, trained only on at least medium correlated features, or trained only on features that are not extracted from the pre-classifier. We therefore chose MLP as our correction classifiers.

Before conducting training on the combined train and dev dataset, we separated the texts for which the RoBERTa pre-classifier had predicted the human label from those for which it had predicted the machine label, thus creating two splits. Then, we trained two separate MLPs on the two splits of the training data using the 26 features identified as relevant in the feature selection process (4.2.2). The idea behind this approach was that the models might learn in which cases the fine-tuned transformer classified the data incorrectly, and would thus have to be corrected. The test data was then prepared by calculating the 26 features, on which the pair of MLP correction classifiers made the final predictions.

model	label	prec.	rec.	f1
fusion model	human	0.85	0.85	**0.85**
	machine	0.86	0.86	**0.86**
accuracy: **0.85**				
pre-classifier	human	**0.99**	0.48	0.64
	machine	0.68	**0.99**	0.81
accuracy: 0.75				
MLP	human	0.53	0.89	0.67
	machine	0.75	0.30	0.43
accuracy: 0.58				

Table 2: Precision, recall, f1-score, and overall accuracy for the submitted fusion model and two models for comparison: the RoBERTa pre-classifier and an MLP model trained with the selected linguistic features. The support for the 'human' class is 16,272 and for the 'machine class 18,000.

5 Results

Table 2 shows the performance of the submitted fusion model, obtained using the `classification_report` from scikit-learn. Overall, the fusion model achieves an accuracy of 85%. The table additionally shows the performances of two other models on the test data in comparison: (i) the RoBERTa pre-classifier; (ii) an MLP model that was trained with the same hyperparameters as used for the correction classifiers and the same features selected in the feature selection process (see Section 4.2.2), except for the ones extracted from the pre-classifier (see Section 4.2.1).

Although the pre-classifier performed fairly well on the dev data (accuracy: 0.89, for more details see Table 6 in the Appendix), we opted for a fusion model with a correction layer in order to improve robustness for data from new generators and domains. The implementation of the two correction MLPs corroborated the hypothethis. On the test data, the accuracy of the pre-classifier drops to 0.75, while the addition of the correction layer improved the accuracy to 0.85.

A closer look at the recall and precision for the two classes 'human' and 'machine' reveals that the fusion model balances out the problems of the pre-classifier and the MLP. The high precision and low recall of the pre-classifier for 'human' and vice versa for 'machine' indicate that it is biased

towards the 'machine' class. Accordingly the relatively high precision and low recall of the MLP classifier for 'machine' and vice versa for 'human' indicate that it is biased towards the 'human' class. In contrast, the fusion model shows equally high precision and recall for both classes.

As described in Section 4.1, the pre-classifier is obtained from the RoBERTa base OpenAI detector by further training. Comparing its performance to the original model (see Table 7, Appendix), fine-tuning has led from a bias towards the 'human' class to a bias towards the 'machine class'. This is likely due to the fact that the fine-tuning data had an imbalance towards machine-generated texts.

5.1 Error Analysis

An error analysis was completed in three parts: We examined the influence of the different labels, the features, and the correctional classifiers on accuracy. The influence of the domain was not examined since there was only one domain present in the test data.

When inspecting the label distribution for the misclassified texts, we can see an almost perfect 50% split between human and machine-labeled texts. Between the models, the errors are not distributed as evenly, as shown in Figure 3 in the Appendix. GPT4 and dolly texts were misclassified most often, followed by Cohere, DaVinci, and chatGPT, while BLOOMz texts were rarely classified incorrectly. Since GPT4 texts were not seen in the train or dev data, it is not unsurprising that those texts were classified least accurately. A further reason could be that GPT4-generated texts are known to be very 'human-like', hence harder to differentiate from human texts.

Figure 2 shows the classification by the pre-classifier and whether it was modified by the correction classifier (the same data in numbers is given in Table 3). A text identified as human by the pre-classifier was typically classified accurately and only rarely adjusted by the correcting classifier. For the texts where the pre-classifier predicted a machine label, the prediction was corrected often. However, as shown in table 3, 2,308 cases should have been corrected and were not. The machine label predictions by the pre-classifier have caused most errors, as that label was predicted so often. This is also reflected in the recall of the pre-classifier-only model in Table 2.

Finally, we correlated all features used in the

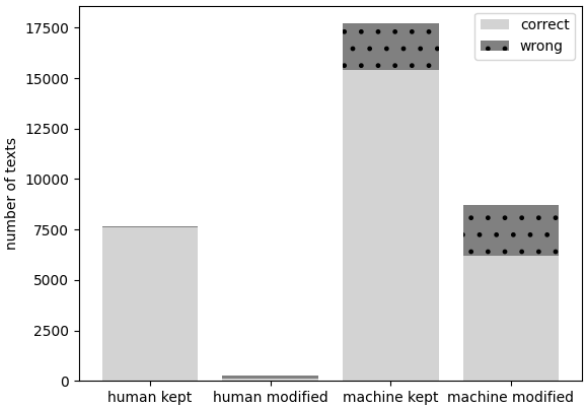


Figure 2: Left two bars: predictions by correction classifier correcting texts pre-classified as "human", right two bars: predictions by correction classifier correcting texts pre-classified as "machine".

pre-classifier	h		m	
correction classifer	h	m	m	h
correct	7622	86	15402	6189
wrong	12	153	2308	2500

Table 3: Classification errors split by prediction by the pre-classifier and correction classifier (h = human, m = machine).

fusion model with the labels that were predicted incorrectly. The strongest correlation was shown by the features *1st UMAP-dimension (Jaccard)* (- 0.82), *ratio of CEFR-B1 words (stem)* (- 0.71), *ratio of CEFR-B2 words (stem)* (- 0.57), *neutral sentiment score* (- 0.51), and *ratio of pronouns to content words* (0.55). Since the correlations are on the wrong predictions, a strong negative correlation indicates a correlation with an incorrectly predicted human label (label 0), while a strong positive correlation implies the opposite. It is possible that the low correlation threshold chosen for feature acceptance led to the inclusion of features initially weakly correlated with the labels in the training and development data, which may have adversely affected the correctional classifiers' decisions. Alternatively, the test set data might exhibit a different distribution for those features compared to the training and development data.

6 Conclusion and Limitations

Overall, this study has highlighted the benefit of using a fusion architecture consisting of a pre-classifier and linguistically informed correctional classifiers. By adding syntactic, stylistic, sentiment, frequency- and word difficulty-based features, we

were able to improve the performance of a fine-tuned pre-trained RoBERTa model for AI generated text detection and adjust the bias towards the machine label. Because our fusion model uses a pre-trained RoBERTa model, all computations for this paper can be run locally or, in the case of the RoBERTa fine-tuning, using a free Google Colab account. This means that our model can be easily expanded and leaves a smaller environmental footprint.

Future studies could expand our fusion model by incorporating more semantic-level or complex features such as contextual predictability, as well as fine-tuning the pre-classifier using more, balanced data. Our code is available on GitHub[11].

References

Nikolay Babakov, David Dale, Ilya Gusev, Irina Krotova, and Alexander Panchenko. 2023. Don't lose the message while paraphrasing: A study on content preserving style transfer. In *Natural Language Processing and Information Systems*, pages 47–61, Cham. Springer Nature Switzerland.

José del Campo-Ávila, Gonzalo Ramos-Jiménez, and Rafael Morales-Bueno. 2007. Incremental learning with multiple classifier systems using correction filters for classification. In *Advances in Intelligent Data Analysis VII*, pages 106–117, Berlin, Heidelberg. Springer Berlin Heidelberg.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Jochen Hartmann. 2022. Emotion english distilroberta-base. `https://huggingface.co/j-hartmann/emotion-english-distilroberta-base/`.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Zhixin Lai, Xuesheng Zhang, and Suiyao Chen. 2024. Adaptive ensembles of fine-tuned transformers for llm-generated text detection. *arXiv preprint arXiv:2403.13335*.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rexhep Shijaku and Ercan Canhasi. 2023. Chatgpt generated text detection.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Superior Court of Gwinnett County. Mark Walters v. OpenAI,L.L.C.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

[11] `https://github.com/ansost/art-nat-HHU-semeval2024`

A Features with medium or high correlation

feature	corr.
1st UMAP-dimension (jaccard)	0.94
logits for label 1 from pre-classifier	0.92
roBERTa prediction	0.92
positive sentiment score	0.28
ratio of determiners to content words	0.28
ratio of pronouns to content words	0.25
score for formal	0.22
ratio of CEFR-B1 words (stem)	0.20
ratio CEFR (all levels) words (stem)	0.17
ratio of CEFR-B2 words (stem)	0.17
ratio of CEFR-A2 words (stem)	0.13
ratio of CEFR-B1 words (lemma)	0.13
ratio of conjunctions to words	0.13
ratio of CEFR-A2 words (lemma)	0.12
score for joy	0.11
ratio of fantasy words	0.10
score for neutral	0.10
ratio of Wikipedia words	0.10
word ratio of top 10% freq. Wiktionary words	0.10
word ratio of top 20% freq. Wiktionary words	0.10
...	
score for fear	−0.10
1st UMAP-dimension (cosine)	−0.10
score for anger	−0.11
ratio of pronouns to words	−0.15
number of hapaxes	−0.17
score for informal	−0.22
ratio of adverbs to words	−0.22
prop. of unfreq. words to content words	−0.25
TTR	−0.27
number of unique words	−0.27
negative sentiment score	−0.28
mean depth of dep. tree for sentences	−0.40
max depth of dependency tree	−0.45
2nd UMAP-dimension (jaccard)	−0.59
logits for label 0 from pre-classifier	−0.92

Table 4: Features with medium or strong positive or negative correlation ($-0.1 \leq \text{corr} \leq 0.1$) with label 1 (machine) in train data

B Fusion model features

feature name
type-to-token ratio (TTR)
ratio of adverbs to content words
ratio of pronouns to content words
ratio of determiners to content words
ratio of conjunctions to content words
ratio CEFR (all levels) words
ratio of CEFR-A2 words
ratio of CEFR-B1 words
ratio of CEFR-B2 words
number of hapaxes
ratio of frequent words to content words
ratio of hapaxes to content words
1st UMAP-dimension (cosine)
negative sentiment score
positive sentiment score
score for anger
score for fear
score for neutral
score for joy
score for formal
score for informal
max depth of dependency tree
mean depth of dependency tree for sentences
word ratio of top 10% freq. Wiktionary words
1st UMAP-dimension (jaccard)
logits for label 0 from RoBERTa pre-classifier

Table 5: Features used to train the fusion model.

C RoBERTa pre-classifier performance on dev

label	precision	recall	f1-score
human	0.91	0.86	0.88
machine	0.87	0.91	0.89

Table 6: Precision, recall, f1-score, and support for the RoBERTa pre-classifier on the dev data. Accuracy is 0.89

D RoBERTa base OpenAI detector performance on test

label	precision	recall	f1-score
human	0.57	**0.98**	0.72
machine	**0.95**	0.34	0.50

Table 7: Precision, recall, f1-score, and support for the RoBERTa base OpenAI detector on the test data. Accuracy is 0.64

E Distribution of errors

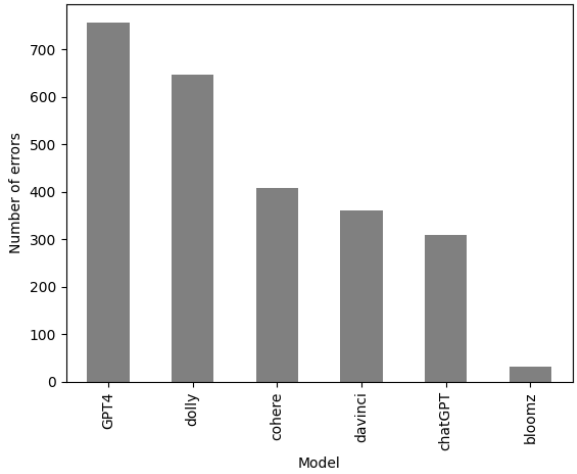


Figure 3: Distribution of models in false predictions.

AIMA at SemEval-2024 Task 3: Simple Yet Powerful Emotion Cause Pair Analysis

Alireza Ghahramani Kure[◇], **Mahshid Dehghani**[◇], **Mohammad Mahdi Abootorabi**[◇], **Nona Ghazizadeh**[◇], **Seyed Arshan Dalili**[◇], **Ehsaneddin Asgari**[§]
[◇] NLP & DH Lab, Computer Engineering Department, Sharif University of Technology
[§] Qatar Computing Research Institute, Doha, Qatar
{a.ghahramani, mahshid.dehghani, mahdi.abootorabi, nona.ghazizadeh, seyedarshan.dalili}@sharif.edu
easgari@hbku.edu.qa

Abstract

The SemEval-2024 Task 3 presents two subtasks focusing on emotion-cause pair extraction within conversational contexts. Subtask 1 revolves around the extraction of textual emotion-cause pairs, where causes are defined and annotated as textual spans within the conversation. Conversely, Subtask 2 extends the analysis to encompass multimodal cues, including language, audio, and vision, acknowledging instances where causes may not be exclusively represented in the textual data. Despite this, our model addresses Subtask 2 using the same architecture as Subtask 1, focusing solely on textual and linguistic cues. Our architecture is organized into three main segments: (i) embedding extraction, (ii) cause-pair extraction & emotion classification, and (iii) post-pair-extraction cause analysis using QA. Our approach, utilizing advanced techniques and task-specific fine-tuning, unravels complex conversational dynamics and identifies causality in emotions. Our team, AIMA (MotoMoto at the leaderboard), demonstrated strong performance in the SemEval-2024 Task 3 competition ranked as the 10^{th} rank in subtask 1 and the 6^{th} in subtask 2 out of 23 teams. The code for our model implementation is available on https://github.com/language-ml/SemEval2024-Task3.

1 Introduction

The task of Emotion-Cause Pair Extraction in Conversations holds significant importance in advancing the field of emotion analysis. Unlike previous endeavors that primarily focused on recognizing emotions, this task delves deeper into understanding the underlying causes behind emotional expressions within conversational contexts (Wang et al., 2023). Recognizing that emotions are conveyed not only through words but also through vocal intonations and facial expressions, the field has shifted towards multimodal emotion recognition. This move aims to understand how emotions are interwoven with text, sound, and visual cues in dialogue (Wang et al., 2023).

The SemEval-2024 Task 3 (Wang et al., 2024, 2023; Xia and Ding, 2019) encompasses two subtasks aimed at extracting emotion-cause pairs in conversational contexts. Subtask 1 focuses on textual emotion-cause pair extraction, where causes are defined and annotated as textual spans within the conversation. In contrast, Subtask 2 broadens the analysis to incorporate multimodal cues, including language, audio, and vision. The task is based on the multimodal conversational emotion cause dataset ECF (Wang et al., 2023). Figure 1 illustrates an example of the task and the annotated dataset.

In this paper, we introduce an approach based on a model architecture consisting of three key components: (i) embedding extraction, (ii) cause-pair extraction & emotion classification, and (iii) cause extraction via QA post-pair detection. Utilizing advanced techniques and fine-tuning on specific datasets, our goal is to dissect complex conversational dynamics and pinpoint nuances that indicate emotional causality.

Although our architecture supports multimodal data—including text, audio, and video through concatenations of the embeddings of these modalities using pretrained models—this study specifically harnesses textual data, as our primary focus is on addressing subtask 1.

2 Related Work

This section provides an overview of two key areas in the field of emotion analysis: Emotion Recognition in Conversation and Emotion-Cause Pair Extraction in Conversations.

Emotion Recognition in Conversation: Emotion recognition in conversation, a burgeoning field, aims to decipher and understand the complex interplay of emotions within dialogues. ERC has seen significant advancements in recent years (Kim

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1698–1703

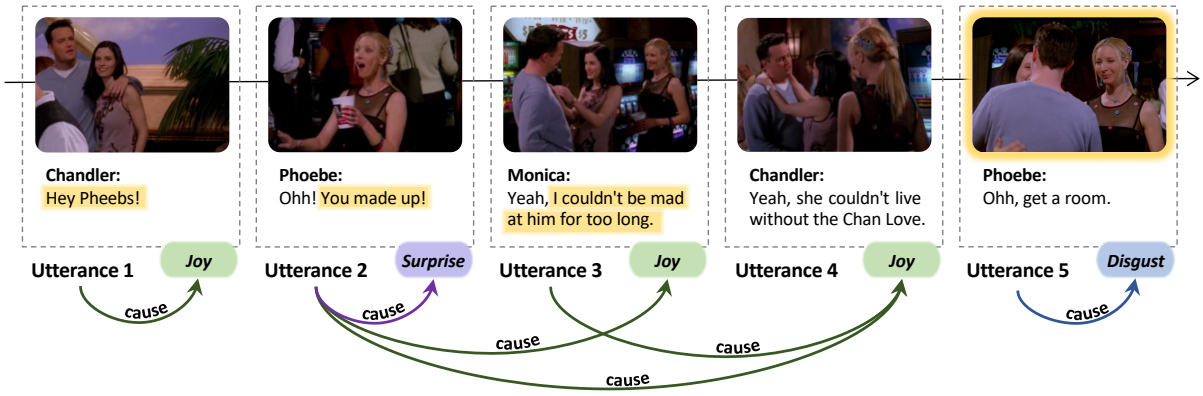


Figure 1: An example of the annotated conversation in ECF (Wang et al., 2023) dataset, illustrating the multimodal nature of emotion causes. Each arc points from the cause utterance to the emotion it triggers. The cause spans have been highlighted in yellow.

and Vossen, 2021; Zheng et al., 2023). These approaches have shown promising results on popular datasets such as IEMOCAP (Busso et al., 2008) and MELD (Poria et al., 2019).

EmoBERTa (Kim and Vossen, 2021) enhances RoBERTa (Liu et al., 2019) for emotion recognition in conversation (ERC) on datasets IEMOCAP (Busso et al., 2008) and MELD (Poria et al., 2019), by incorporating speaker information and dialogue context. It preprocesses dialogues, representing them as sequences with speaker annotations and context segments. EmoBERTa extends RoBERTa to handle multiple segments and utilizes a linear layer with softmax nonlinearity for sequence classification.

The FacialMMT (Zheng et al., 2023) framework comprises two key stages. Initially, a pipeline method is employed to isolate the face sequence of the real speaker within each utterance. Following this, a multi-modal facial expression-aware emotion recognition model is applied. This model utilizes frame-level facial emotion distributions and incorporates multi-task learning to improve utterance-level emotion recognition. Experimental evaluations conducted on the MELD (Poria et al., 2019) dataset validate the effectiveness of FacialMMT.

Emotion-Cause Pair Extraction in Conversations: The task of Emotion-Cause Pair Extraction in Conversations is pivotal for advancing our understanding of the nuanced interplay between emotions and their underlying triggers within dialogues, offering insights into human communication, cognition, and interpersonal dynamics.

The paper (Wang et al., 2023) introduces a baseline system, MC-ECPE-2steps, comprising two steps. Firstly, it employs multi-task learning to extract emotions and causes separately, utilizing word-level encoding and utterance-level encoders to derive representations specific to each. Secondly, it combines the predicted emotions and causes into pairs and employs BiLSTM and attention mechanisms to obtain pair representations. Subsequently, non-causal pairs are filtered out using a feed-forward neural network. Additionally, the system incorporates multimodal features from text, audio, and video modalities to enhance the extraction process. In addition to this approach, there exist other methodologies for Emotion-Cause Pair Extraction in Conversations (Xia and Ding, 2019; Zheng et al., 2022), some of which leverage question answering techniques (Nguyen and Nguyen, 2023).

3 System Overview

Our model architecture, illustrated in Figure 2, is designed with the capacity to incorporate a diverse set of inputs from various sources such as text, video, and audio to perform emotion-cause analysis within conversational contexts. However, for the purpose of addressing subtask 1, we specifically utilized textual data.

Embedding Extraction and Emotion Classification: In the Embedding Extraction phase, we leverage the EmoBERTa (Kim and Vossen, 2021) model specifically designed for text embedding. EmoBERTa's selection is based on its proven effectiveness in capturing the nuanced emotional dynamics inherent in conversational data, thereby facilitating precise emotion classification of the utterances.

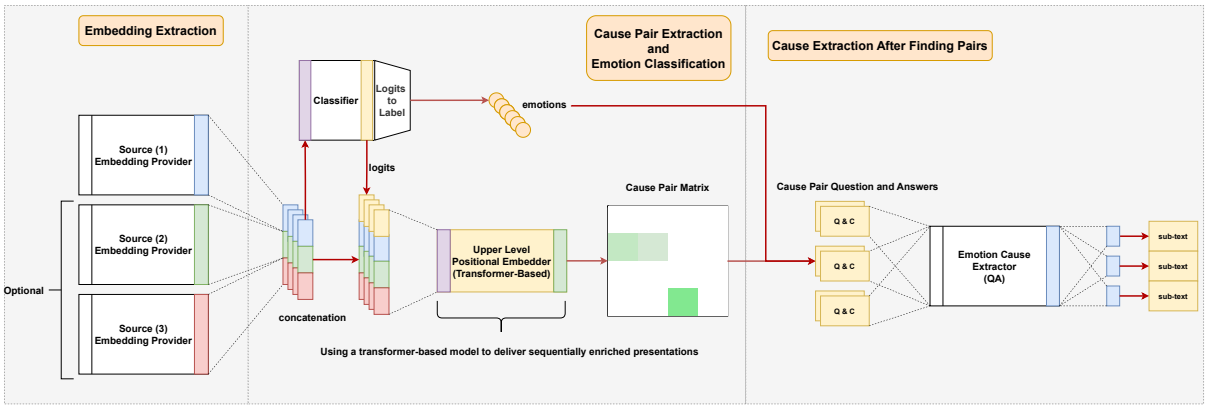


Figure 2: The schema of our proposed model for emotion-cause analysis, meticulously partitioned into three core segments: **Embedding Extraction**, **Cause Pair Extraction and Emotion Classification**, and **Cause Extraction After Finding Pairs**

Additionally, it's noteworthy that EmoBERTa's emotion classification schema encompasses classes such as "neutral, joy, surprise, anger, sadness, disgust, and fear," mirroring the emotion categories present in the task dataset. This alignment ensures consistency in emotion classification across datasets. Moreover, we fine-tune EmoBERTa on the task dataset, further enhancing its ability to capture emotion-specific nuances within conversational utterances. Notably, the original model (before fine-tuning) achieves an accuracy of 67% on the training data, indicating a good performance in emotion classification.

Causality Matrix Extraction: The embeddings of utterances, combined with logits from the classification task, are processed by a Transformer-based Encoder. This includes positional embeddings added to input vectors and a sequence of transformer encoder layers. The model's output, derived from the attention weights of the final layer, forms a causality matrix. This matrix highlights potential causal relationships within dialogue utterances, capturing the complex dynamics of conversation. The approach enriches data with emotion-specific insights, streamlining the identification of diverse emotion classes directly within the embeddings. In the following, the process of extracting the causality matrix is explained in detail.

Causality Matrix Extraction Process:

1. Initial combination of embeddings and logits:

$$combined = [s_1, s_2, s_3, logits] \quad (1)$$

where s_1, s_2, and s_3 are embeddings for an utterance, and $logits$ are the output from the classification model M_c, computed as $logits = M_c([s_1, s_2, s_3])$.

2. Application of dropout and addition of positional embeddings:

$$input = dropout(combined) + e_{pos} \quad (2)$$

Here, e_{pos} represents the positional embeddings, which are added to the dropout-modified combined inputs to incorporate positional information into the sequence representation. Specifically, e_{pos} encodes the position of each utterance within the conversation, enriching the model's understanding of dialogue structure and the sequential context of each utterance.

3. Generation of the causality matrix through the transformer encoder layers:

$$C_m = A_N(l_{1:N-1}^{encoder}(input)) \quad (3)$$

Here, $l_i^{encoder}$ denotes the i-th transformer encoder layer, with $N-1$ indicating that the input sequentially passes through all layers up to the $N-1$-th layer. A_N refers to the attention weights from of the N-th (last) encoder layer. The causality matrix, C_m, is specifically derived from these attention weights applied to the output of the $N-1$-th layer, which has been processed by all preceding encoder layers and enhanced with positional embeddings. This matrix captures the causal interactions within the dialogue, as inferred from the attention mechanism of the transformer's final layer.

Question Generation for Causality Pairs: Following the emotion classification task, where emotions within the dialogue are identified, a causality matrix is created. For each emotion-cause pair detected in this matrix, the system generates a structured query to facilitate the extraction of the causal text segment. The prompt, constructed only for these detected pairs, follows the template:

"Which part of the text {target_utterance} is the reason for {speaker}'s feeling of {emotion} when {main_utterance} is said?"

The Cause Extraction After Finding Pairs phase utilizes a question-answering model to interrogate the text, pinpointing exact sub-texts that substantiate the identified emotional triggers. (see Figure 3).

This study undertook a thorough evaluation of various question-answering (QA) models, uncovering areas where each model could be enhanced. Among the models examined, DistilBERT (Sanh et al., 2019) and BERT (Devlin et al., 2018) showed considerable promise for application within our research framework. Ultimately, we selected the deepset/deberta-v3-base-squad2, a pre-trained QA model, for our specific task requirements. This choice was informed by the model's foundation on the DeBERTa-v3-base architecture (He et al., 2021) and its prior fine-tuning on the SQuAD2 dataset (Rajpurkar et al., 2016), which includes both answerable and unanswerable questions. By further fine-tuning this model on our dataset, we ensured its proficiency in accurately extracting causal text segments from conversational contexts, a critical capability for our emotion-cause analysis.

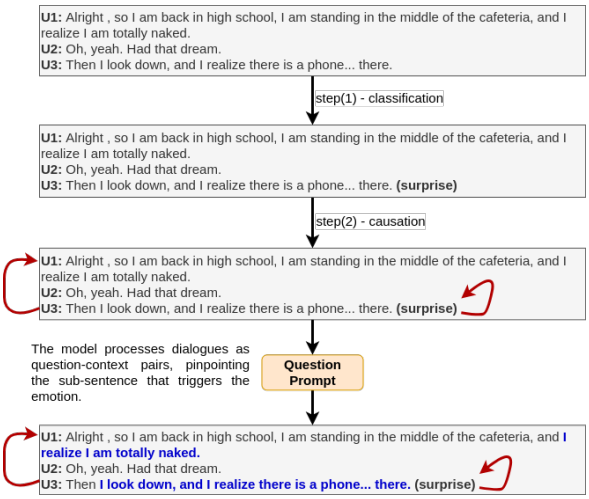


Figure 3: An example of the model's question-answering mechanism in action. After classifying emotions in the dialogue and creating the causality matrix, a question prompt is generated only for detected emotion-cause pairs. This diagram demonstrates the process of identifying the causative segment within the dialogue that led to the emotional response, with the causative text being highlighted in the context of the detected pairs.

4 Experimental Setup

4.1 Dataset Preparation

Dataset Preparation for Attention Model: The dataset preparation for cause pair extraction and emotion classification procedure commenced with the loading of conversation data and emotion-cause pairs, accompanied by preprocessing steps tailored for model training. A custom dataset class facilitated the loading and processing of data, extracting essential details like conversation ID, utterances, and emotion-cause pairs. Subsequently, a collate function was employed to organize individual samples into batches suitable for model input, focusing solely on text and generating attention targets based on the presence of cause pairs within the textual data.

Dataset Preparation for QA Model: The dataset preparation for subtext emotion cause extraction using question answering involved constructing samples for question answering by generating questions and contexts solely from text data. Each sample comprised a question formulated with a predefined prompt, the context concatenating all utterances from the conversation, and the answer containing the cause subtext. The dataset then underwent preprocessing to train the question-answering model, utilizing a pre-trained tokenizer to align tokenized inputs with the original text and determine the start and end positions of the answers within the textual context.

4.2 Training

Training the Attention Model: The attention model was optimized using mean squared error loss and the AdamW optimizer with a learning rate of 1e-4.

Training the QA Model: The QA model was trained over 25 epochs with a batch size of 8.

4.3 Evaluation Metrics

Our models' performance was gauged using F1 scores across the six primary emotion categories, with additional emphasis on weighted averages to account for class imbalances. Subtask 1 evaluations incorporated both Strict Match and Proportional

Match metrics to assess the accuracy of textual span identification for emotional causes.

Metric	Strict	Proportional	Weighted
Precision	0.0217	0.2018	0.2779
Recall	0.0217	0.2081	0.2486
F1-Score	0.0217	0.2049	0.2584

Table 1: Performance metrics for team AIMA (MotoMoto) in SemEval-2024 Task 3.

5 Results

5.1 Quantitative Findings

Our team, MotoMoto, participated in the SemEval-2024 Task 3 competition and secured the 10th rank in Subtask 1 and 5th rank in Subtask 2. The official metrics for our team's performance are as shown in Table 1 To explore the effectiveness of our approach, we compare it with the MC-ECPE-2steps (Wang et al., 2023) method, which represents our baseline. The comparison is based on the weighted average F1 scores achieved by both approaches, as presented in Table 2.

Approach	Weighted-average F1
MC-ECPE-2steps	0.3000
-Audio	0.2764
-Video	0.2993
-Audio - Video	0.2625
Ours	0.2584

Table 2: Comparison of Approaches with Baselines based on Weighted Average F1

5.2 Error Analysis

Our investigation into the discrepancies between our system's predictions and the ground truth leveraged the detailed insights from the confusion matrix (Table 3). The analysis underscores our emotion classification module's exceptional performance, notably in accurately identifying 'Neutral' and 'Joy' emotions with 4400 and 1576 correct instances, respectively. This substantiates our model's adeptness at recognizing emotions within conversations. Despite these strengths, the emotion-cause pair extraction component displayed variations, such as over or under-identification of causes compared to the ground truth annotations. Nevertheless, the precision of our model in identifying correct causes, as highlighted by specific successes in the confusion matrix, confirms its effectiveness in discerning emotions. These observations suggest that while our model excels in accurately identifying emotions, there is a valuable opportunity to refine the identification of causal factors within conversations for further improvement.

Table 3: Confusion Matrix for 13,619 dialogues. The model demonstrates no signs of overfitting, hence the entire train dataset is utilized to report this table.

	Neutral	Joy	Surprise	Anger	Sadness	Disgust	Fear
Neutral	4400	610	242	218	307	31	121
Joy	392	1576	136	82	70	19	26
Surprise	154	134	1380	77	34	17	44
Anger	168	180	192	823	88	71	93
Sadness	203	79	82	94	581	29	79
Disgust	83	34	41	77	25	143	11
Fear	70	36	42	24	35	8	158

6 Conclusion

Our investigation into emotion-cause pair extraction presents a paradigm shift towards simplicity and efficiency without compromising performance. By adopting a streamlined approach, we have demonstrated that high-impact emotion analysis does not necessarily require heavy computational resources or complex multimodal data integration. Our participation in the SemEval-2024 Task 3 competition has validated our methodology, securing commendable rankings and highlighting the efficacy of our model. The results underscore the potential of cost-effective solutions in the realm of emotion analysis, opening doors to wider applicability in resource-constrained environments. Looking forward, we aim to further optimize our model's efficiency and explore the integration of lightweight multimodal data processing techniques. This endeavor not only reinforces the viability of minimalist approaches but also sets a new benchmark for future research in emotion-cause analysis.

References

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of

deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Taewoon Kim and Piek Vossen. 2021. Emoberta: Speaker-aware emotion recognition in conversation with roberta.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Huu-Hiep Nguyen and Minh-Tien Nguyen. 2023. Emotion-cause pair extraction as question answering.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^{2}Workshop*.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Wenjie Zheng, Jianfei Yu, Rui Xia, and Shijin Wang. 2023. A facial expression-aware multimodal multi-task learning framework for emotion recognition in multi-party conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15445–15459.

Xiaopeng Zheng, Zhiyue Liu, Zizhen Zhang, Zhaoyang Wang, and Jiahai Wang. 2022. UECA-prompt: Universal prompt for emotion cause analysis. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7031–7041, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

AIMA at SemEval-2024 Task 10: History-Based Emotion Recognition in Hindi-English Code-Mixed Conversations

Mohammad Mahdi Abootorabi[◇], **Nona Ghazizadeh**[◇], **Seyed Arshan Dalili**[◇], **Alireza Ghahramani Kure**[◇], **Mahshid Dehghani**[◇], **Ehsaneddin Asgari**[§]
[◇] NLP & DH Lab, Computer Engineering Department, Sharif University of Technology
[§] Qatar Computing Research Institute, Doha, Qatar
{mahdi.abootorabi, nona.ghazizadeh, seyedarshan.dalili, a.ghahramani, mahshid.dehghani}@sharif.edu
easgari@hbku.edu.qa

Abstract

In this study, we introduce a solution to the SemEval 2024 Task 10 on subtask 1, dedicated to Emotion Recognition in Conversation (ERC) in code-mixed Hindi-English conversations. ERC in code-mixed conversations presents unique challenges, as existing models are typically trained on monolingual datasets and may not perform well on code-mixed data. To address this, we propose a series of models that incorporate both the previous and future context of the current utterance, as well as the sequential information of the conversation. To facilitate the processing of code-mixed data, we developed a Hinglish-to-English translation pipeline to translate the code-mixed conversations into English. We designed four different base models, each utilizing powerful pre-trained encoders to extract features from the input but with varying architectures. By ensembling all of these models, we developed a final model that outperforms all other baselines.

1 Introduction

The first subtask of SemEval 2024 Task 10 focuses on Emotion Recognition in Conversation (ERC) (Kumar et al., 2023). This subtask requires the design of a model capable of predicting an emotion for each utterance. Our final system is an ensemble of four high-performing models we developed in this paper. Our primary strategy involves leveraging powerful pre-trained models and utilizing the context of preceding and succeeding utterances in the conversation. We also consider the sequential information of the conversation to accurately predict emotions. The final system is designed to work with Hindi-English code-mixed conversations. A detailed description of the task is available in (Kumar et al., 2024).

ERC is an emerging research frontier in Natural Language Processing (NLP), that aims to identify emotions in conversational data. The ability to accurately recognize emotions in conversation is crucial for a variety of applications, including opinion mining from social media platforms (Poria et al., 2019). ERC is also extremely important for generating emotion-aware dialogues that require an understanding of the user's emotions. It is useful in various sectors, such as healthcare for psychological analysis and education to aid in understanding student frustration (Antony et al., 2021).

ERC presents several research challenges due to the complexity and rapid changeability of emotions in conversation. The same words can convey different emotions depending on the context, adding a layer of complexity to the task (Kumar et al., 2023). This complexity is further amplified in code-mixed conversations, a common phenomenon in multilingual societies and online social media platforms where two or more languages are used interchangeably. The challenges in ERC for code-mixed conversations include **(i) Linguistic Complexity**, due to complex linguistic structures and sentence or word-level language switches; **(ii) Insufficient Training Data**, as the scarcity of annotated datasets hampers the training of deep learning models; **(iii) Cultural Nuances**, since emotions can be expressed differently across cultures and languages; and **(iv) Ambiguity and Context-Dependence**, as word meaning and emotions vary based on context and language.

2 Background

The official dataset for this task is the MaSaC dataset (Bedi et al., 2023), a mixed Hindi-English language dataset relevant to our study of emotion recognition in code-mixed dialogues (Kumar et al., 2023). This dataset consists of approximately 8506 train, 1354 validation, and 1580 test sentences. ERC has garnered significant attention in the NLP community due to its potential applications.

Recent research in ERC has attempted to ad-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1704–1710

dress these challenges, but there are still many areas for improvement. For instance, most current approaches to ERC focus on text-based data, overlooking the rich emotional information that can be gleaned from other modalities such as voice tone and facial expressions (Kumar et al., 2023).

With recent advances in Large Language Models (LLMs), many works leverage the power of these large models for ERC task (Tu et al., 2023). (Lei et al., 2023) introduced a novel approach, which leverages LLMs to reformulate the ERC task from a discriminative framework to a generative one. This approach has shown significant improvements over previous models on several ERC datasets. While considerable research has focused on discerning the emotions of individual speakers in monolingual dialogues, understanding the emotional dynamics in code-mixed conversations has received relatively less attention. This is the gap our study aims to address. (Kumar et al., 2023) proposed an innovative approach that integrates commonsense information with dialogue context to interpret emotions more accurately in code-mixed dialogues. They developed a pipeline based on a knowledge graph to extract relevant commonsense facts and fuse them with the dialogue representation. (Wadhawan and Aggarwal, 2021), the closest work to ours, introduced a new Hinglish dataset for emotion detection in Hindi-English code-mixed tweets. They trained various deep learning approaches, including transformer-based models, for emotion recognition task performance on this dataset.

This paper aims to delve deeper into the current state of ERC and propose models to take a step toward solving these challenges. Our method differs from existing approaches in that it incorporates both context and sequential information to improve emotion prediction performance.

3 System Overview

3.1 Preprocessing Data

In the preprocessing stage, we implement a two-step translation process due to the unique nature of our data, which comprises Hindi-English mixed conversations. At present, there are no robust models trained specifically in this mixed language, nor are there translators capable of directly translating Hindi-English mixed text to English with acceptable performance. As a result, we first need to translate our data to English. In the first step, we transform our Hindi-English mixed conversations into Hindi using the indic-trans transliteration module (Bhat et al., 2015), a tool proficient in cross-transliteration among all Indian languages. Following this, we employ SeamlessM4T Medium (Communication et al., 2023) to translate these Hindi conversations into English. The English conversations obtained from this process serve as our preprocessed data.

3.2 Model Architecture

In this section, we propose the model architectures that were used to construct our final ensemble model. We designed three distinct architectures, and the final model is an ensemble of four models trained based on these architectures. The second model follows the same architecture as the first, but it is trained on an augmented dataset. Our system predicts the emotion of the current sentence using majority voting based on the predicted emotions from four base models.

Given the specific domain of the task and the limited number of samples in the dataset, it is crucial to strike a balance between model complexity and performance. Overly complex models may lead to overfitting, especially given the unique distribution of our dataset. Conversely, overly simple models may not capture the complexity of this particular task. Therefore, we aimed to find a balance, ensuring adequate model complexity to learn effectively from the data without leading to overfitting. Furthermore, due to the limited dataset for this task and the special domain and emotions that are used, such as contempt, we leveraged the encoder component of a pre-trained RoBERTa-based model (Liu et al., 2019) for the emotion recognition task in sentences, and fine-tuned it for our specific task and domain. This model was trained on the GoEmotions dataset (Demszky et al., 2020), allowing us to employ the capabilities of pre-trained models for our task. This encoder was incorporated into all of our base models for sentence encoding. In the following parts, each of our base models is explained in detail. An overview of architectures is shown in Figure 1.

3.2.1 Simple History-Based Model

This model leverages both the current sentence, for which we aim to predict the emotion, and the preceding sentence along with its associated emotion as historical information to enhance the model's prediction. Both the current and previous sentences are processed through our pre-trained encoder to

obtain their respective embeddings. We then employed a multi-head attention mechanism (Vaswani et al., 2017) with 8 heads. In this mechanism, the keys are derived from the embeddings of the previous sentence, while the queries and values are derived from the current sentence. The use of 8 heads in the attention mechanism enables the model to capture information from different representational spaces at various levels of abstraction. This design allows the model to focus on the most relevant parts of the current sentence based on the context provided by the previous sentence.

For emotion representation, we utilized a 50-dimensional embedding space learned by our model. The embedding of the previous emotion and the output of the attention mechanism are concatenated and passed through a feed-forward classifier to predict the emotion. This classifier consists of two linear layers, a LeakyReLU activation function, and a Softmax layer for output normalization.

3.2.2 Simple History-Based Model + Data Augmentation

This model architecture is identical to the base model described earlier. The key difference lies in the training data. We used a Pegasus paraphrase model (Zhang et al., 2019) to augment our dataset and increase its size. We expanded our dataset by randomly selecting three sentences from the first ten paraphrases of each original sentence. Given the limited size of the original dataset, this augmentation method should enhance the model's learning capability by exposing it to a wider range of data.

3.2.3 Full History-Based Model

This model, which is an extension of the Simple History-Based model, aims to leverage more historical information for improved performance. In addition to the current sentence, previous sentence, and previous emotion, we also incorporated the concatenated string of all previous sentences in the conversation into our model. The rationale behind this is to enable the model to access additional information and gain a better understanding of the context of the current sentence within the conversation. The concatenated string of all previous sentences is processed through our pre-trained encoder to obtain the history embedding. This encoding is then passed through a simple feed-forward neural network, which consists of two linear layers, a batch normalization layer, a dropout layer, and a LeakyReLU activation function. This network transforms the 768-dimensional input into a 128-dimensional space.

The processing for the current sentence, previous sentence, and previous emotion remains the same as in the Simple History-Based Model. For the classifier network, we concatenated the output of the feed-forward network for previous sentences with the output of the attention mechanism and the emotion embedding. This concatenated vector is then passed to the classifier to predict the current emotion. The classifier comprises three linear layers, a batch normalization layer, two dropout layers, a LeakyReLU activation function, a ReLU activation function, and a Softmax layer for output normalization.

3.2.4 Context-Aware GRU-Based Model

This model, more complex than its predecessors, introduces several key modifications. Firstly, it incorporates information from both the preceding and succeeding sentences in a conversation, allowing the model to leverage both past and future contexts. Secondly, in contrast to previous architectures that use the emotion of the previous sentence, this model omits this feature to prevent error propagation during the inference phase. If a model incorrectly predicts the emotion of one sentence, it could potentially use this incorrect information when predicting the emotion of the next sentence, leading to further errors. Lastly, this model employs a Gated Recurrent Unit (GRU) (Chung et al., 2014; Cho et al., 2014), enabling it to leverage the sequential information in the conversation.

The model processes all sentences up to and including the current one (for which we want to predict the emotion) and the next sentence through our pre-trained encoder to obtain their embeddings. If the current sentence in the conversation has more than three previous sentences, only the last three are considered, making the model focus on the most recent context. These embeddings are then passed through a stacked GRU, consisting of two GRUs with a hidden dimension of 256 and a dropout rate of 0.25. Both the current and next sentences went through a transformation via a linear layer and a dropout layer to generate output encodings in a common 256-dimensional space. The last two hidden layers of the GRU are concatenated and passed through a multi-head self-attention mechanism, similar to our previous models.

The output of the last layer of the GRU, the output of the attention mechanism, and the trans-

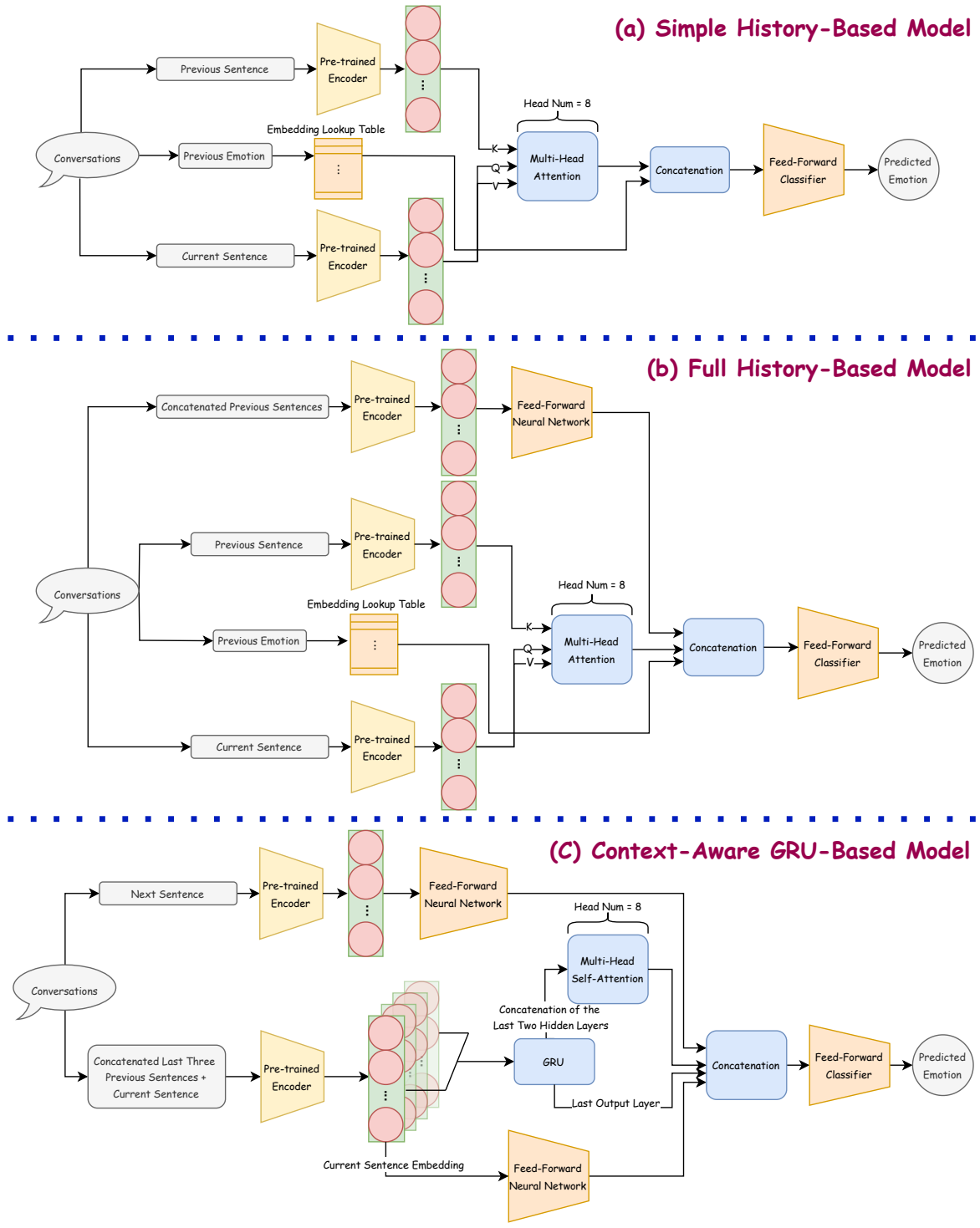


Figure 1: Three proposed base model architectures for predicting the emotion of the current sentence. (a): This model utilizes only the basic historical information from the conversation. (b): This model leverages information from all past sentences, in addition to the information used in the previous architecture. (c): This model employs GRU to leverage sequential information and incorporates future information to gain a more comprehensive understanding of the context of the current sentence.

Model Name	Weighted F1	Accuracy	Weighted Precision	Weighted Recall
GPT-3.5 Turbo	0.2662	0.3070	0.2582	0.3070
Decision Tree	0.2895	0.2937	0.2900	0.2937
Linear Regression	0.3394	0.4405	0.4117	0.4405
Fine-tuned Sentence Emotion Recognition	0.3683	0.4506	0.3667	0.4506
Simple History-Based Model	0.4018	0.4380	0.4043	0.4380
Simple History-Based Model + Data Augmentation	0.3780	0.4253	0.3712	0.4253
Full History-Based Model	0.3992	0.4285	0.3963	0.4285
Context-Aware GRU-Based Model	0.4058	0.4373	0.4024	0.4373
Final Model (Ensemble)	**0.4080**	**0.4430**	**0.4090**	**0.4430**

Table 1: Performance of various models on the test dataset. The first group of models represents the baselines. The second group consists of models based on our proposed architectures. The final model, an ensemble of four proposed models, represents the performance of our final system.

formed outputs of the current and next sentences are concatenated and passed to a feed-forward classifier to predict the current emotion. The classifier comprises two linear layers, a dropout layer, a LeakyReLU activation function, and a Softmax layer for output normalization.

4 Experimental Setup

We utilized the official dataset provided for the task as the only data source for our system. The default split provided for the task was also used. During the development phase, the validation set was exclusively used for evaluating various steps and experimental configurations. For the final submission, models were fine-tuned on both the training and validation splits. For evaluation purposes, our primary metric was Weighted F1. However, to provide a more comprehensive analysis, we also reported three additional metrics, as detailed in Table 1. Our training process primarily employed the PyTorch and Transformers libraries. All base models were trained using the early stopping method and the AdamW (Loshchilov and Hutter, 2019) optimizer. A learning rate scheduler was used, with a lower learning rate set for the pre-trained encoder (5e-6) compared to other parameters (1e-4). The batch size was set to 1 for the Context-Aware GRU-Based model and 4 for other models during training. The cross-entropy loss function was used for the training.

5 Results

Table 1 presents SubTask 1 results. We compare our approach with four baseline models. The first baseline is GPT 3.5 Turbo, for which we used its API key to input the entire conversation and predict the emotion for each sentence. The results of this baseline model illustrate that this task is much more challenging than general sentence emotion recognition because it is domain-specific. The next two models are traditional ones, namely Linear Regression and Decision Tree, that utilize embeddings extracted from the LaBSE sentence encoder (Feng et al., 2022). The LaBSE model serves as a powerful encoder for our text data, enabling us to achieve comprehensive and multilingual text embeddings. The final baseline model is similar to a Simple History-Based model. It employs our pre-trained encoder but does not use any context, such as the previous sentence, and relies solely on the current sentence.

Moving on to the comparison of our models, we first consider the Simple History-Based model. By comparing its results with the Full History-Based model, we find that most of the information for predicting the emotion is contained in the current and the previous sentence. Therefore, information from all of the previous sentences is not as useful for predicting emotion. Our second model, which uses data augmentation, does not perform well. This is likely due to overfitting and the domain-specific nature of the conversations, making data augmentation less effective. As can be seen in our models, the Context-Aware GRU-Based model outperforms the others. This is because it incorporates information from both the preceding and succeeding sentences and the GRU can leverage the sequential information in the conversation. The closeness of

the results between the Context-Aware GRU-Based model and the Simple History-Based model reinforces our assumption that most information for predicting emotion is in the current and previous sentence. All of our models outperform the baselines. For our final model, we create an ensemble of these four models using majority voting. This ensemble model outperforms each individual model, achieving an F1-score of 0.4080.

6 Conclusion

In this paper, we proposed a novel method to address the Code-Mixed Emotion Recognition in Conversations (ERC) challenge. Our approach leverages the power of pre-trained large models and incorporates both previous and future context information of the current utterance, as well as sequential information of the conversation up to that point, to recognize each utterance's emotion. In addition to our primary model, we utilized other base models with different architectures based on various Deep Learning components to tackle this problem. By ensembling all of these models, we developed a final system that outperforms previous models.

Despite these advancements, Code-Mixed ERC remains a challenging task with significant potential for further investigation. Future research directions could include designing robust encoders capable of processing code-mixed dialogues and predicting emotions in an end-to-end manner. Moreover, collecting more data on these code-mixed dialogues is necessary to improve the performance of models. Furthermore, we can explore more complex models that incorporate different information from various modalities to achieve better performance. This work serves as a stepping stone towards more sophisticated emotion recognition systems for code-mixed dialogues.

References

Col Cj Antony, B Pariyath, Siti Noorfatimah Safar, Azharuddin Sahil, and Nair Ar. 2021. Emotion recognition-based mental healthcare chat-bots: A survey. *SSRN Electronic Journal*.

M. Bedi, S. Kumar, M. Akhtar, and T. Chakraborty. 2023. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Transactions on Affective Computing*, 14(02):1363–1375.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, Bapi Akula, Peng-Jen Chen, Naji El Hachem, Brian Ellis, Gabriel Mejia Gonzalez, Justin Haaheim, Prangthip Hansanti, Russ Howes, Bernie Huang, Min-Jae Hwang, Hirofumi Inaguma, Somya Jain, Elahe Kalbassi, Amanda Kallet, Ilia Kulikov, Janice Lam, Daniel Li, Xutai Ma, Ruslan Mavlyutov, Benjamin Peloquin, Mohamed Ramadan, Abinesh Ramakrishnan, Anna Sun, Kevin Tran, Tuan Tran, Igor Tufanov, Vish Vogeti, Carleigh Wood, Yilin Yang, Bokai Yu, Pierre Andrews, Can Balioglu, Marta R. Costa-jussà, Onur Celebi, Maha Elbayad, Cynthia Gao, Francisco Guzmán, Justine Kao, Ann Lee, Alexandre Mourachko, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Paden Tomasello, Changhan Wang, Jeff Wang, and Skyler Wang. 2023. Seamlessm4t: Massively multilingual multimodal machine translation.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances.

Geng Tu, Bin Liang, Bing Qin, Kam-Fai Wong, and Ruifeng Xu. 2023. An empirical study on multiple knowledge from ChatGPT for emotion recognition in conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12160–12173, Singapore. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in Hindi-English code-mixed data: A transformer based approach. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 195–202, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Team MGTD4ADL at SemEval-2024 Task 8: Leveraging (Sentence) Transformer Models with Contrastive Learning for Identifying Machine-Generated Text

Huixin Chen[1], Jan Büssing[2], David Rügamer[2,3], Ercong Nie[†1,3]
[1] Center for Information and Language Processing (CIS), LMU Munich,
[2] Institute for Statistics, LMU Munich,
[3] Munich Center for Machine Learning (MCML)
{chen.huixin, jan.buessing}@campus.lmu.de
david.ruegamer@stat.uni-muenchen.de
nie@cis.lmu.de

Abstract

This paper outlines our approach to SemEval-2024 Task 8 (Subtask B), which focuses on discerning machine-generated text from human-written content, while also identifying the text sources, i.e., from which Large Language Model (LLM) the target text is generated. Our detection system uses Transformer-based techniques and incorporates various pre-trained language models (PLMs), which are tools that help understand and process language, including sentence transformer models. Additionally, we incorporate Contrastive Learning (CL) into the classifier to improve the detecting capabilities and employ Data Augmentation methods. Ultimately, our system achieves a peak accuracy of 76.96% on the test set of the competition, configured using a sentence transformer model integrated with CL methodology.

1 Introduction

The emergence of sophisticated Large Language Models (LLMs) has significantly blurred the lines between human-written and machine-generated texts, prompting an urgent need for systems capable of accurately distinguishing between the diverse sources.

In response, our team participated in SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection, as defined by Wang et al. (2024). This task aims to identify the origin of texts across various languages and domains, addressing critical concerns around the misuse of LLMs. We focused on Subtask B, which involves classifying English texts by their generative sources. This task adopted a fine-grained label set, for distinguishing not only between human-written and machine-generated texts, but also among texts generated by different machines. Our system leveraged Transformer-based pre-trained language models (PLMs) as well as its variant, Sentence Transformer models (Reimers and Gurevych, 2019). By applying Contrastive Learning (CL) approaches, which aimed at enhancing model robustness and generalization to our system, our best approach yielded a modest improvement over the baseline on the test set, achieving an accuracy of 76.96% compared to the baseline's 74.61%, and ranking 20th in the competition. The code for our system, detailed further in this paper, is made available at: https://github.com/banjuessing/adl_semeval24_mgtd.

2 Background and Related Work

The introduction of the M4 dataset by Wang et al. (2023) offers a comprehensive landscape for evaluating detection techniques across various generators, domains, and languages. The research done on the M4 dataset underscores the difficulties in generalizing detection across different domains and generators, highlighting the limitations of current approaches.

Data for Subtask B of SemEval-2024 Task 8, focusing on the detection of human-written against machine-generated texts from multiple generators across monolingual (English) contexts, is derived from the original M4 dataset. The dataset comprises 71,027 training and 3,000 development/test samples, distributed across multiple sources — Wikipedia, Reddit, arXiv, and wikiHow — with the testing data focused on the out-of-domain PeerRead domain. The task demands the identification of text origins, whether human or machine-generated by models. This underscores the necessity for systems that are adept at handling multi-class and out-of-domain classification challenges. In response to these challenges, our approach builds upon the insights from prior work. Abdalla et al. (2023), for instance, applied linguistic- and transformer-based method to detecting the author-

[†] Corresponding author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1711–1718

ship of text. We also considered the methods used in the M4 dataset paper to compare with.

3 System Overview

Having outlined the urgency and relevance of distinguishing machine-generated text, we now describe our Transformer-based approach to tackle this issue. Our system tackles machine-generated text detection by carefully selecting a suite of transformer models (RoBERTa$_{\text{BASE}}$, RoBERTa$_{\text{LARGE}}$ (Liu et al., 2019), GPT-2 Small (Radford et al., 2019), XLNet-Base (Yang et al., 2019)), sentence transformer models (all-mpnet-base-v1[1], all-mpnet-base-v2[2], all-roberta-large-v1[3]), and integrating two different Contrastive Learning techniques, namely Supervised Contrastive Learning (SCL) (Gunel et al., 2020; Khosla et al., 2020) and Dual Contrastive Learning (DualCL) (Chen et al., 2022), alongside data augmentation strategies, inspired by (Bhattacharjee et al., 2023). Initially, we conducted hyperparameter tuning across both transformer and sentence transformer models in their base forms with trivial cross-entropy (CE) loss to identify optimal configurations. Subsequently, we refined our model selection to GPT-2 Small, RoBERTa$_{\text{BASE}}$, RoBERTa$_{\text{LARGE}}$, and all-roberta-large-v1, based on performance metrics on the enriched test set, which is described in section 4.1, further experimenting with combination of contrastive learning technique variants to enhance detection accuracy. Our approach culminates in the additional application of data augmentations aiming to improve robustness and generalizability.

3.1 Transformers

We began our exploration with a diverse set of transformer models: RoBERTa$_{\text{BASE}}$, RoBERTa$_{\text{LARGE}}$, GPT-2 Small and XLNet-Base, distinguished by unique architectural designs and pre-training objectives. RoBERTa, as an encoder model enhanced with optimized training approach dynamic masking strategy on longer sequences, offers robustness and depth in understanding context. GPT-2 with its generative capabilities and autoregressive training objective, provides insights into the sequence prediction dynamics often employed by text generation models. XLNet, incorporating permutation-based training, may capture bidirectional context and outperform traditional unidirectional models in understanding complex sentence structures. The different characteristics of those models grant us a comparative edge in detecting generated content. We conducted extensive hyperparameter tuning to identify the configurations that yield optimal performance on the classification task, which was crucial for ensuring that each model was leveraged to its fullest potential.

3.2 Sentence Transformers

In parallel, we evaluated three sentence transformer models: all-mpnet-base-v1, all-mpnet-base-v2, and all-roberta-large-v1. The decision to incorporate sentence transformers alongside traditional transformers was driven by their further pre-training on sentence pairs for generating semantically rich embeddings (Reimers and Gurevych, 2019), which potentially offers a more nuanced understanding of the essence of entire sentences. The selection of the three variants was informed by their pre-training paradigms and underlying architectures, which may influence their performance on text classification tasks. The all-mpnet-base models with a relatively smaller model size of 420 MB and hidden dimension of 768, derived from the MPNet (Song et al., 2020), are notable for their optimized permuted language modeling pre-training upon XLNet. The distinction between v1 and v2 primarily lies in the maximum sequence length, with v1 having 512 tokens and v2 having 384 tokens. The all-roberta-large-v1 model, on the other hand, is built upon the RoBERTa architecture with a larger model size of 1360 MB, a larger hidden dimension of 1024 but a smaller context window size of 256 tokens. Similar to the transformer models, hyperparameter tuning was performed to fine-tune these models for our specific task, ensuring that the models' configurations were optimized.

3.3 Contrastive Learning

Based on the initial evaluations, we narrowed our focus to GPT-2 Small, RoBERTa$_{\text{BASE}}$, RoBERTa$_{\text{LARGE}}$, and all-roberta-large-v1. These models were subjected to further experiments to test the efficacy of Contrastive Learning methods in enhancing their performances.

Driven by the training objectives of the sentence transformers and the intuition that in the embedding space, examples from the same source tend to

[1]https://huggingface.co/sentence-transformers/all-mpnet-base-v1

[2]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[3]https://huggingface.co/sentence-transformers/all-roberta-large-v1

be grouped together, while examples from different generators or human could be potentially pushed apart to be distinguished, we integrated SCL loss and DualCL loss with our selected models. Both loss functions utilize Contrastive Learning in the supervised setting. Following Gunel et al. (2020), the SCL loss directly takes the samples from the same class as positive samples and the samples from different classes as negative samples, while the DualCL loss simultaneously learns from the features of input samples $\mathcal{L}_z$ and the parameters of classifiers $\mathcal{L}_\theta$ in the same space with label-aware data augmentation (Chen et al., 2022). The overall loss that we used to optimise the models is then one of the two following combinations of two losses, where the λ adjusts the balance between the primary loss function and the contrastive loss component:

$$\mathcal{L}_{overall}^{SCL} = (1-\lambda)\mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{SCL}} \quad (1)$$

$$\mathcal{L}_{overall}^{DualCL} = (1-\lambda)\mathcal{L}_{\text{CE}} + \frac{1}{2}\lambda\mathcal{L}_{\text{DualCL}} \quad (2)$$

where $\mathcal{L}_{\text{DualCL}} = \mathcal{L}_z + \mathcal{L}_\theta$.

Each model was trained and evaluated using one of these Contrastive Learning methods, in addition to the traditional CE loss, to compare their effectiveness systematically. Hyperparameter tuning was again employed for each combination of model and loss to ensure optimal settings.

3.4 Hyperparameter Optimization

For hyperparameter optimization (HPO) within our detection system, we employed a grid search strategy. Specifically, when training our models using the conventional CE loss, our tuning focused solely on optimizing the learning rate of the optimizer. Conversely, in scenarios where models were trained with the incorporation of SCL loss or DualCL loss, we extended our tuning efforts to include both the learning rate and the λ value.

3.5 Data Augmentation

Finally, we investigated the role of data augmentation in further enhancing the models' ability to discern machine-generated text. Selecting the top-performing model configurations from the previous steps, we applied various data augmentation techniques using `nlpaug` library[4] (Ma, 2019), including synonym replacement and random word swap to enrich the training dataset. This step aimed to introduce variability and complexity to the training process, testing the hypothesis that augmented data could lead to better generalization and robustness.

[4]https://github.com/makcedward/nlpaug

4 Experimental Setup

4.1 Data

The dataset for SemEval-2024 Task 8 encompasses a broad spectrum of text generators, encompassing both human-authored and machine-generated sources. The machine generated texts include outputs from advanced LLMs: BLOOMz, ChatGPT, Cohere, Davinci-003, and Dolly-v2. The data features diverse domains, including arXiv, WikiHow, Wikipedia, Reddit and PeerRead. This composition challenges us to distinguish human-written text from machine-generated content and further identify the specific LLM responsible.

The dataset was strategically split by organizers to promote an out-of-domain testing scenario, with the test set solely containing PeerRead texts absent from training data, comprising only 500 samples evenly distributed across each generator. This limitation led us to enrich our test dataset by incorporating all available samples from the original M4 dataset specific to the PeerRead domain, thereby aiming for a comprehensive analysis within our experimental framework. Utilizing the full PeerRead dataset provided us access to 14,566 data points, significantly enhancing our ability to conduct a deeper exploration of text detection capabilities. The distribution of data points across each model/source is as follows:

Model/Generator	Number of Samples
BLOOMz	2,334
ChatGPT	2,344
Cohere	2,342
Davinci-003	2,344
Dolly-v2	2,344
Human	2,858

Table 1: Distribution of data points across models/sources for the Peerread domain in our enriched test dataset.

To address the requirements of our experimental setup, we partitioned the original training dataset, as provided by the organizers, into two subsets: 90% for training and 10% for validation, where the labels and source domains of the samples are evenly distributed. This division was consistently applied

across all experiments to maintain uniformity in the evaluation process. Additionally, the enriched test dataset, as previously described, was employed as the test dataset for all experimental validations.

4.2 Hyperparameters

Under the experimental setup, a consistent approach was adopted for hyperparameter selection across all models to ensure comparability of the results. We ultilized AdamW (Loshchilov and Hutter, 2017) optimizer with a default weight decay of 0.01 for training each model across all experiments. Training was conducted with mixed precision for 20 epochs, incorporating an early stopping mechanism triggered by 3 consecutive epochs of loss increase. For the hyperparameter tuning of all transformer models and sentence transformer models using CE loss, a grid search methodology was implemented. The learning rate parameters explored were {1e-5, 2e-5, 5e-5}, with the exception of the RoBERTa$_{\text{LARGE}}$ model, for which a range of {1e-6, 2e-6, 5e-6} was tested. When integrating either SCL loss or DualCL loss, We explored the learning rate combined with λ values of {0.02, 0.1, 0.2} to adjust the influence of contrastive loss. Our decisions of selecting best performed models based on the accuracy of each model's performance on our enriched test dataset.

5 Results and Discussions

In our analysis of the performance of four transformer models as our baseline on the task of detecting machine-generated text, distinct variations in accuracy underscore the impact of model design and size on effectiveness, as shown in Table 2. The GPT-2 Small model, achieving the highest accuracy at 73.25%, outperformed both XLNet-Base and RoBERTa models. This superior performance could be attributed to GPT-2's architecture, primarily designed as a decoder model for generating text, which may inherently provide it with a nuanced capability to distinguish between human and machine-generated texts. When comparing models within the same family, RoBERTa$_{\text{BASE}}$'s performance surpasses that of RoBERTa$_{\text{LARGE}}$. This observation suggests that increasing model size, and thereby complexity, does not necessarily translate to better performance in detecting machine-generated text and a smaller model might be more effective than its larger counterpart. This could be due to the diminishing returns of model capacity expansion in this specific task.

Model	Accuracy
XLNet-Base	64.22
GPT-2 Small	**73.25**
RoBERTa$_{\text{BASE}}$	67.31
RoBERTa$_{\text{LARGE}}$	64.29

Table 2: Accuracy of the transformer models on the enriched test set. The results are reported as the best performance among each model's hyperparameter configurations.

In our evaluation of three sentence transformer models, we observed distinct performance outcomes that offer insights into the influence of model architecture and input sequence length on accuracy, as shown in Table 3. Specifically, the all-mpnet-base-v1 and all-mpnet-base-v2, which share the same foundational model and architectural parameters including model size and hidden dimension, demonstrated only a marginal difference in accuracy (61.36% for v1 and 60.73% for v2). This slight discrepancy in performance, despite v1's capability to process longer input sequences than v2, suggests that an extended context window does not inherently guarantee superior detection efficacy in our task. Conversely, the all-roberta-large-v1 model, characterized by its robust architecture and a higher hidden dimension of 1024, although with a reduced context window size, markedly outperformed the aforementioned models, achieving an accuracy of 69.96%. This outcome underscores the observation that a larger context window, contrary to expectations, may not be as critical for enhancing machine-generated text detection as previously assumed.

Model	Accuracy
all-mpnet-base-v1	61.36
all-mpnet-base-v2	60.73
all-roberta-large-v1	**69.96**

Table 3: Accuracy of the sentence transformer models on the enriched test set. The results are reported as the best performance among each model's hyperparameter configurations.

Our explorations with selected best models from previous experiments further led to insightful observations regarding the performance of incorporating contrastive learning methods, as shown in Table 4. For GPT-2 Small model, both the CL losses corrupted the performance, indicating the

alignments of CL losses may not suitable for a decoder model. For the RoBERTa$_{BASE}$ model, integrating CL methodologies yielded results comparable to those obtained using traditional CE loss with a slight underperformance. Similarly, the RoBERTa$_{LARGE}$ model, when augmented with CL methods, demonstrated only a marginal improvement under 2% over the conventional CE loss approach. Conversely, the all-roberta-large-v1 sentence transformer model showed a strong contrast in performance when leveraging two contrastive learning losses. The model variant with additional SCL loss markedly outperformed the accuracy achieved with standard CE loss, resulting in the best model variant across all our experiments. However, incorporating DualCL loss resulted in substantially poorer performance compared to the baseline, hinting at potential mismatches between the DualCL objective and the sentence transformer model for the task-specific requirements. Upon comparing the overall performances, the all-roberta-large-v1 model outperformed remarkably both the RoBERTa$_{BASE}$ and RoBERTa$_{LARGE}$ models, indicating that the adaptation and specialization of sentence transformers significantly contribute to discerning the subtle intricacies of machine-generated texts with the SCL loss further enhancing this ability.

Model	Loss	Accuracy (%)
GPT-2 Small	CE	73.25
	CE+SCL	72.53
	CE+DualCL	58.15
RoBERTa$_{BASE}$	CE	67.31
	CE+SCL	66.85
	CE+DualCL	66.64
RoBERTa$_{LARGE}$	CE	64.29
	CE+SCL	64.94
	CE+DualCL	65.94
all-roberta-large-v1	CE	69.96
	CE+SCL	**74.60**
	CE+DualCL	53.16

Table 4: Accuracy of the selected best performed models with various loss functions on the enriched test set. The results are reported as the best performance among each combination's hyperparameter configurations.

Incorporating data augmentation techniques, as detailed in section 3.5, to further train the GPT-2 Small and all-roberta-large-v1 model, which demonstrated top-2 performances in previous experiments, resulted in a significant decrease in performance, details shown in Table 5 in Appendix A.1. This decline was observed across both configurations of utilizing CE loss and the combination of CE loss and SCL loss, despite their initially high accuracy on the enriched test set. A potential reason for this downturn could be the introduction of noise or irrelevant variations through data augmentation, which may have led to the models' reduced ability to generalize from the augmented data, ultimately detracting from its capability to accurately distinguish machine-generated texts.

As we analyse model performance dynamics, an intriguing pattern of overfitting emerged among some of the top-performing model configurations. Upon testing earlier checkpoints of these models against the enriched test set, it was observed that certain pre-final checkpoints exhibited superior performance compared to the final models, which had achieved the highest validation accuracy. This phenomenon suggests that models slightly earlier in their training phase, before reaching peak validation accuracy, may generalize better to unseen data when detecting the machine generated texts. We report the detailed performance dynamics in Table 6 in Appendix A.2.

As we observe our best model's performance on the enriched test dataset for each generator, we find that the model demonstrates a robust ability to accurately identify texts generated by Dolly-v2, BLOOMz, ChatGPT, and Cohere, indicating a strong alignment with the characteristics prevalent in the outputs from these sources. However, it encounters significant challenges when attempting to classify texts originating from human authors and the Davinci-003 model. We report the detailed confusion matrix in Appendix A.3. This insight points to the need for further model refinement and training to bridge the gap in detection capabilities across some certain text origins.

6 Conclusion

In conclusion, our paper presents a comprehensive approach to SemEval-2024 Task 8 (Subtask B), focusing on the detection of machine-generated text and its attribution to specific Large Language Models (LLMs). Leveraging Transformer-based methods, pre-trained language models (PLMs), Contrastive Learning (CL), and Data Augmentation techniques, we have developed a robust detection system achieving a peak accuracy of 74.69%. Our findings underscore the effectiveness of integrat-

ing CL into the classification process and highlight the strength of leveraging diverse PLMs for improved performance in discerning between human and machine-generated text.

Acknowledgments

We thank Ercong Nie for his valuable guidance and support in our participation in the SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection as part of Applied Deep Learning Course at the Ludwig-Maximilians-Universität München organized by Prof. David Rügamer.

References

Mohamed Hesham Ibrahim Abdalla, Simon Malberg, Daryna Dementieva, Edoardo Mosca, and Georg Groh. 2023. A benchmark dataset to distinguish human-written and machine-generated scientific papers. *Information*, 14(10).

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. Conda: Contrastive domain adaptation for ai-generated text detection.

Qianben Chen, Richong Zhang, Yaowei Zheng, and Yongyi Mao. 2022. Dual contrastive learning: Text classification via label-aware data augmentation. *CoRR*, abs/2201.08702.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *CoRR*, abs/2011.01403.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *CoRR*, abs/2004.11362.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *CoRR*, abs/2004.09297.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

A Appendix

A.1 Models with Data Augmentation

Table 5 shows the detailed result of experiments implemented with Data Augmentation methods.

A.2 Model Performances Dynamics

Table 6 shows the training dynamics.

A.3 Confusion Matrix of the Best Performed Model

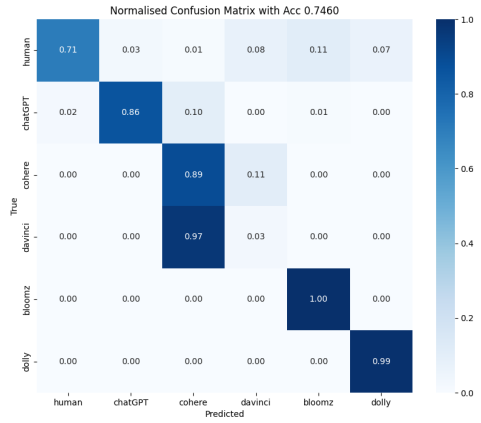


Figure 1: Confusion matrix of the our best model's (all-roberta-large-v1 with CE+SCL loss) performance on the enriched test dataset described in section 4.1 with texts only in Peerread domain.

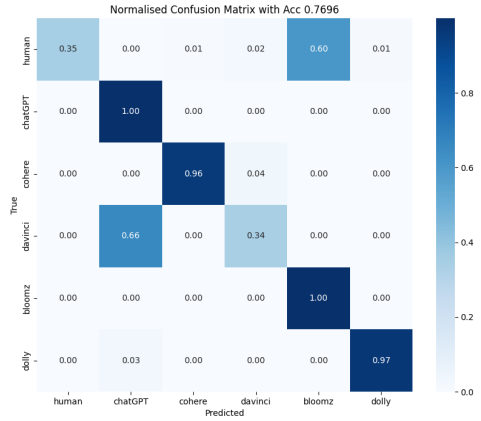


Figure 2: Confusion matrix of the our best model's (all-roberta-large-v1 with CE+SCL loss) performance on the test dataset that provided by the organizers.

Model	Loss	Augmentation	Accuracy (%)
GPT-2 Small	CE	No	73.25
		Yes	56.82
all-roberta-large-v1	CE+SCL	No	74.60
		Yes	58.81

Table 5: Accuracy of the best-performed GPT-2 Small and all-roberta-large-v1 model with various loss functions and data augmentation on the enriched test set. The results are reported as the best performance among each combination's hyperparameter configurations.

Save Point (epoch)	1	2	3	4	5	6	7	8	9	10
$\text{RoBERTa}_{\text{BASE}}$	65.34	66.04	-	58.51	**66.42**	65.47	-	-	63.50	
all-roberta-large-v1	66.15	69.57	70.93	73.05	-	-	**74.60**	-	-	70.84
GPT-2 Small	69.93	-	71.25	-	-	**72.53**				

Table 6: Accuracy(%) of the three selected model configurations' performances across different epochs on the enriched test set. We select $\text{RoBERTa}_{\text{BASE}}$ with DualCL(λ=0.02), all-roberta-large-v1 with SCL(λ=0.2) and GPT-2 Small with SCL(λ=0.2) to observe the performance dynamics, because among the best performed model configurations they have long enough converge processes. The dashes in the table indicate no model checkpoint is saved in that epoch due to no increase in the validation accuracy. Saved model checkpoints in the later epochs have higher validation accuracy.

ClusterCore at SemEval-2024 Task 7: Few Shot Prompting With Large Language Models for Numeral-Aware Headline Generation

Monika Singh, Sujit Kumar, Tanveen and Sanasam Ranbir Singh
Indian Institute of Technology Guwahati
{s.monika, sujitkumar, t.tanveen , ranbir}@iitg.ac.in

Abstract

The generation of headlines, a crucial aspect of abstractive summarization, aims to compress an entire article into a concise, single line of text despite the effectiveness of modern encoder-decoder models for text generation and summarization tasks. The encoder-decoder model commonly faces challenges in accurately generating numerical content within headlines. This study empirically explored LLMs for numeral-aware headline generation and proposed few-shot prompting with LLMs for numeral-aware headline generations. Experiments conducted on the *NumHG* dataset and NumEval-2024 test set suggest that fine-tuning LLMs on *NumHG* dataset enhances the performance of LLMs for numeral aware headline generation. Furthermore, few-shot prompting with LLMs surpassed the performance of fine-tuned LLMs for numeral-aware headline generation.

1 Introduction

News articles are one of the most important sources of information in everyday life. News headlines are vital in selecting which news seems relevant to read. As delineated in studies (Wei and Wan, 2017; Gabielkov et al., 2016), headlines play a significant role in making news viral on social media and influencing readers' opinions (Tannenbaum, 1953). Inaccurate, incongruent or misinformation headlines also lead to the spread of misinformation and disinformation over digital platforms (Chesney et al., 2017; Kumar et al., 2022, 2023). Consequently, generating an accurate headline for a news body is essential. Therefore, ensuring the accuracy of headlines is essential for maintaining the credibility and usefulness of news publications. The task of headline generation, which is a form of text summarization, aims to condense a lengthy source text into a concise summary. This summary, typically presented as a headline, encapsulates the main points of the original text, providing readers with a quick overview of the content (Huang et al., 2023).

In earlier studies on headline generation, various sequence-to-sequence and encoder-decoder methods have been employed to extract relevant headlines from news articles (Nallapati et al., 2016; Chen et al., 2020; Paulus et al., 2018; Song et al., 2019). However, encoder-decoder methods faced challenges in processing large sequences of text. To address these limitations, recent studies (Radford et al., 2018; Devlin et al., 2018; Lewis et al., 2019; Liu et al., 2019; Raffel et al., 2020) have proposed transformer-based models for headline generation by summarizing news articles. While transformer-based models have indeed showcased enhanced capabilities in handling longer text sequences and have exhibited promising outcomes in headline generation tasks; it is noteworthy that their performance in numeral-aware headline generation tasks need to be consistently superior. Despite their overall advancements, transformer-based models may face challenges in accurately incorporating and representing numeric information within generated headlines. Motivated by such observations, the study (Huang et al., 2023) proposed numeral aware headline generation datasets.

This paper introduces our proposed approach and provides a comprehensive analysis of the task of *Numeral-Aware Headline Generation* (Task 3 (2)). Our proposed methodology leverages Few-shot prompting with LLMs, which involves applying few-shot learning techniques to large language models (LLMs) for numeral-aware headline generation tasks. We conduct our experiments using the NumHG dataset (Huang et al., 2023) and the test set provided by the organizer of NumEval Task-3(2). Our experimental results suggest that few-shot prompting-based methods with LLMs are efficient for numeral-aware headline generation.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1719–1726

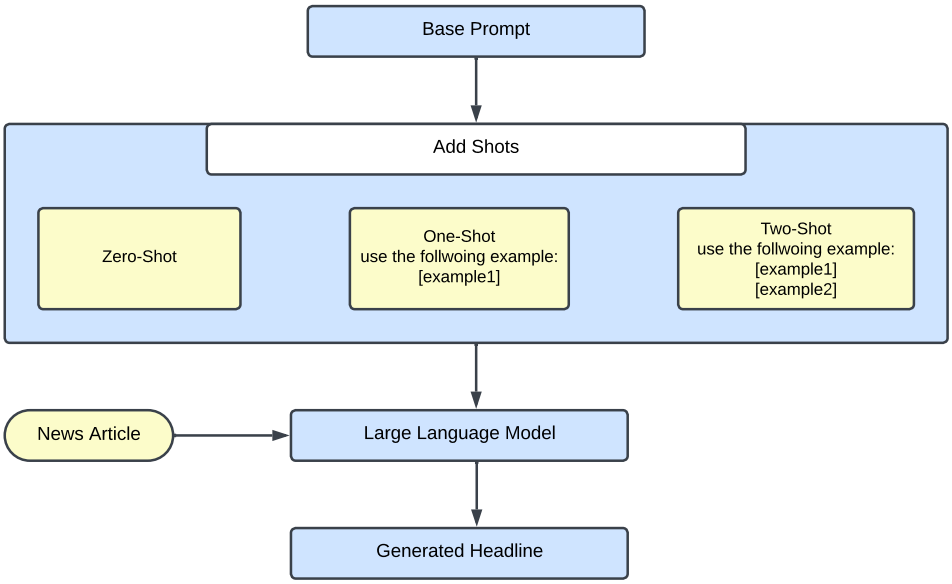


Figure 1: Working diagram of the proposed method.

2 Related Work

Headline generation, a type of text summarization, condenses lengthy source text into a brief summary, usually presented as a headline. This summary captures the main points of the original text, offering readers a quick overview (Huang et al., 2023). Summarization involves extractive and abstractive methods: Extractive selects key sentences, while abstractive generates novel summaries. In prior research investigating headline generation, a range of sequence-to-sequence and encoder-decoder approaches were employed to derive relevant headlines from news articles (Nallapati et al., 2016; Chen et al., 2020; Paulus et al., 2018; Song et al., 2019). However, these approaches encountered challenges, particularly in processing lengthy text sequences. The limitations of encoder-decoder methods in handling large sequences of text hindered their effectiveness in accurately summarizing news articles. To address these shortcomings and enhance the capability of headline generation models, recent research has focused on developing transformer-based architectures (Radford et al., 2018; Devlin et al., 2018; Lewis et al., 2019; Liu et al., 2019; Raffel et al., 2020). Similarly, Large Language Models LLMs such as GPT (Radford et al., 2019), BART (Lewis et al., 2020), T5 (Raffel et al., 2020) and LLaMA (Touvron et al., 2023) have also shown promising state-of-the-art models performance for text generation and summarization task.

Most studies above emphasize word selection and sentence structure, overlooking the significance of accurate numeric values in news headlines. Addressing this gap in the literature, a study (Huang et al., 2023) introduced numeral-aware headline generation datasets to facilitate the development of numeral-aware headline generation methods. Considering the superior performance of Large Language Models (LLMs) in text generation and summarization tasks (Basyal and Sanghvi, 2023), this study conducts an empirical study of LLMs for numeral-aware headline generation. Additionally, an error analysis is performed on the responses of LLMs for numeral aware headline generation. Subsequently, we propose Few-shot prompting with Large Language Models (LLMs) for numeral-aware headline generation.

3 Proposed Method

As mentioned above, the paper aims to study the effect of two important aspects of numeral aware headline generations. First, we study the effectiveness of large language models (LLMs) for numeral-aware headline generations. Second, we propose a few prompting-based methods for numeral-aware headline generations.

3.1 Large Language Models (LLMs):

Considering the effectiveness of LLMs in text summarization (Basyal and Sanghvi, 2023) and headline generations task (Ding et al., 2023). We fine-

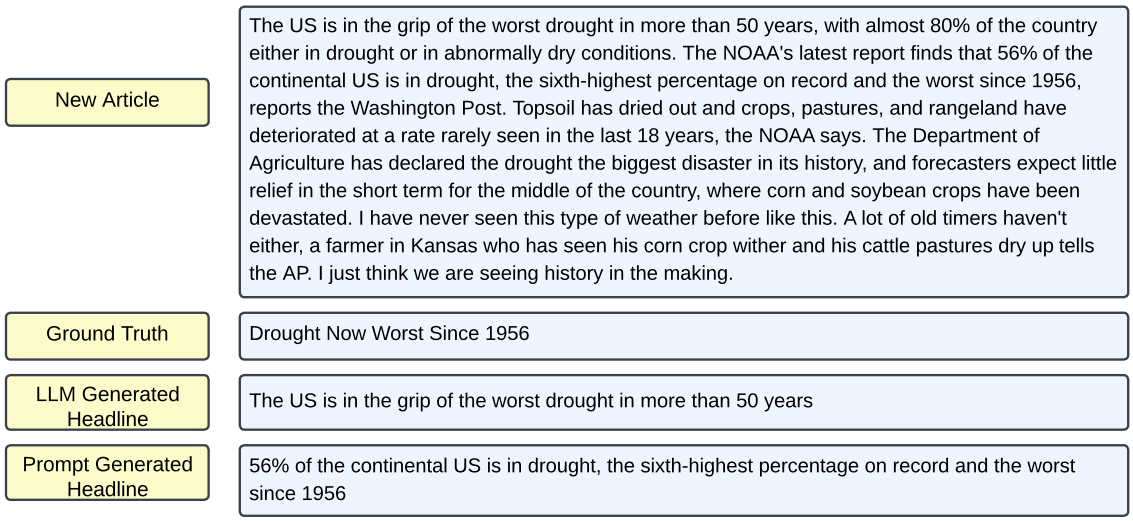


Figure 2: Presents an example comparison of a headline generated by a fine-tuned $T5$ model and a headline generated by a $T5$ model with three shot prompt

tune RoBERTa[1] (Rothe et al., 2020), *Generative Pre-trained Transformer* (GPT-2)[2] (Radford et al., 2019), *Bidirectional and Auto-Regressive Transformers* **BART**[3] (Lewis et al., 2020) and ***T**ext-**T**o-**T**ext **T**ransfer **T**ransformer* **T5**[4] (Raffel et al., 2020) for numeral aware headline generations.

3.2 Few Shot Prompting:

In-context learning denotes a methodology whereby language models acquire proficiency in tasks by utilizing a limited number of examples provided as demonstrations (Dong et al., 2022). The utilization of shot prompting guides the model's output. This approach encompasses three distinct strategies: zero-shot, one-shot, and few-shot prompting. Zero-shot prompting, also called direct prompting, entails assigning a task to the model without providing specific examples, relying solely on the knowledge the model has gained through its training. In contrast, one-shot and few-shot prompting involve presenting examples or 'shots' to the model during runtime, which are references for the expected response's structure or context (Reynolds and McDonell, 2021). The model then utilizes these examples to perform the task. Because these examples are presented in natural language, they offer an accessible method for interacting with language models and facilitate the integration of human knowledge into these models through demonstrations and templates. As evidenced by the findings of several recent studies (van Zandvoort et al., 2023; Schick and Schütze, 2021; Luo et al., 2022), the integration of few-shot learning techniques coupled with prompt instructions has demonstrated a noteworthy enhancement in the quality of text generated or summarized by large language models (LLMs). These observations underscore the potential effectiveness of leveraging few-shot learning methodologies alongside prompt guidance to augment the capabilities of LLMs in generating text of higher quality and relevance. Motivated by such observations regarding few-shot learning with quick text generation and summarization instructions, this study proposes few-shot and prompt engineering-based methods for numeral-aware headline generations. Figure 1 presents the working diagram of our few shot prompting with LLMs-based numeral aware headline generation method. There are three key components of our proposed method, namely:

1. **Few Shot:** We explore three distinct strategies of few-shot prompting: zero-shot, one-shot, and few-shot prompting. These strategies encompass varying degrees of example provision to guide the model's output, allowing for a comprehensive analysis of their respective efficacy in facilitating model performance across different tasks. We have used three examples for methods in our study, which will

[1] https://huggingface.co/google/roberta2roberta_L-24_gigaword

[2] https://huggingface.co/MU-NLPC/CzeGPT-2_headline_generator

[3] facebook/bart-large-cnnÂůHuggingFace

[4] https://huggingface.co/Michau/t5-base-en-generate-headline

be considered three-shot prompting.

2. **Base Prompt:** Here, we provide instruction to a model which guides the model in numeral-aware headline generations. Below is one example of prompt instruction we provided to LLMs for generating numeral-aware headlines.
   Prompt (P1) : `Generate a short headline for a given news article. The headline should be concise and small but represent the content of the news body. The headline may contain a number that could be obtained by performing simple arithmetic operations like addition, subtraction, division, and multiplication or obtained by copying the same valid number from the news article if required to summarize the article.`

3. **Large Language models (LLMs):** This study considers three prominent large language models: GPT (Radford et al., 2018), T5 (Raffel et al., 2020), and LLaMA [5] (Touvron et al., 2023). These models generate headlines that accurately represent given news bodies, utilizing input consisting of the news body itself, prompt instructions, and a few-shot example.

4 Experimental Results and Discussions

4.1 Dataset

We consider the NumHG dataset curated by study (Huang et al., 2023) for training models and the test set provided by *NumEval* organizers for evaluating models. Table 4 presents the characteristics of experimental datasets.

4.2 Experimental Setups

This study incorporates several performance evaluation metrics to assess the effectiveness of models, namely *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE)[6] (Lin, 2004), BERTScore[7] (Zhang* et al., 2020), MoverScore(Zhao et al., 2019) and Num Acc. (Huang et al., 2023) as performance evaluation metrics to evaluate the performance of models. These metrics provide comprehensive insights into various aspects of model performance, including linguistic quality, content overlap, semantic similarity, and numeral accuracy, respectively. Table 3 presents the details of experimental hyperparameters. To replicate the findings in this work, visit GitHub `https://github.com/MONIKASINGH16999/ClusterCore_SemEval2024Task7` to access our code repository.

4.3 Results and discussion

Table 1 illustrates the performance metrics of large language models (LLMs) across various configurations, including *Pretrained*, *Fine-tuned*, and *Shot Prompting*, evaluated on a designated test dataset. This evaluation aims to provide insights into the efficacy and adaptability of LLMs in different settings for numeral-aware headline generation. Initially, we examine the response of LLMs in both the *Pretrained* and *Fine-tuned* setups for numeral-aware headline generation. From Table 1, it is evident that the $T5$ model consistently outperforms the $RoBERTa$, GPT, and $BART$ models across the test dataset in both the *Pretrained* and *Fine-tuned* setups. From such observations, we can claim that the $T5$ model is more suitable for the headline generation tasks compared to $RoBERTa$, GPT, and $BART$. Referring to Table 1, it becomes apparent that fine-tuning these models over the training set enhances their performance and headline generation capability. Subsequently, we curate a subset of the dataset consisting of fifty news headlines and corresponding news bodies. This subset is formed by selecting pairs from the validation dataset where the presence of numeral figures in the headline is deemed particularly significant in accurately representing the content of the news body. Upon manual inspection of the news headlines generated by fine-tuned $RoBERTa$, GPT, $BART$, and $T5$ models over the subset of the dataset comprising fifty samples, our observations suggest that while the generated headlines are contextually similar to the ground truth headlines and effectively represent the content of the news body, the accuracy in representing numeral figures is notably average. From these observations, we can conclude that fine-tuned $RoBERTa$, GPT, $BART$, and $T5$ models exhibit high efficiency in headline generation but display slightly lower efficiency in numeral-

[5] LLaMA
[6] `https://huggingface.co/spaces/evaluate-metric/rouge`
[7] `https://huggingface.co/spaces/evaluate-metric/bertscore`

Table 1: presents the performance of the models over test datasets

	Model	Num Acc.			ROUGE			BERTScore			MoverScore
		Overall	Copy	Reasoning	1	2	L	P	R	F1.	
Pretrained	**RoBERTa**	20.761	31.943	9.579	18.558	10.325	17.394	83.611	84.728	84.158	53.258
	GPT	24.028	34.529	11.527	18.596	12.356	16.879	81.192	76.925	79.058	54.217
	BART	24.137	35.529	12.746	15.7	11.72	14.846	84.264	84.382	84.323	55.321
	T5	23.988	35.995	11.982	**19.023**	9.365	**17.152**	**85.985**	**85.355**	**85.638**	**57.298**
Finetuned	**RoBERTa**	21.726	32.594	10.859	18.558	10.325	17.394	85.5	86.355	85.907	54.258
	GPT	23.265	34.952	11.578	31.896	14.256	29.854	86.935	81.325	84.13	55.941
	BART	25.623	35.621	13.291	32.64	13.587	30.466	86.435	88.324	87.377	57.689
	T5	36.985	37.514	12.852	**34.352**	**13.876**	**32.365**	**87.383**	**89.682**	**88.532**	**59.982**
Shot Prompting	**GPT**	37.259	37.594	12.589	31.746	12.653	29.356	87.659	86.926	87.292	54.989
	T5	37.569	37.295	12.958	30.245	10.941	29.596	89.111	86.922	87.988	58.364
	LLaMA	**38.233**	**38.233**	**13.942**	**37.985**	**14.854**	**33.592**	**90.359**	**89.856**	**90.107**	**59.983**

Table 2: Presents the human evaluation of headlines generated by our proposed system (few shot prompting with LLMs) submitted to NumEval-224. The organizer of NumEval-2024 does this human evaluation of generated headlines.

Submission	Num Acc. (50 Headlines)	Recommendation (100 News)
ClusterCore	**1.60**	**31**
Noot Noot	1.68	11
Infrrd.ai	**1.81**	**22**
np$_{problem}$	1.57	14
hinoki	1.67	16
Challenges	1.70	10
NCL$_{NLP}$	1.73	16
YNU-HPCC	1.69	15
NoNameTeam	1.59	12

Table 3: Details of Experimental Setups and Hyperparameters

Hyperparameters	**Value**
Batch Size	16
Learning Rate	0.01
Maximum #word in news body	250
Maximum #word in headline	15

aware headline generation. One possible reason behind this discrepancy could be the requirement for complex mathematical reasoning capabilities in numeral-aware headline generation tasks. To enhance the performance of models in numeral-aware headline generation tasks, this study employs shot prompting methods. Shot prompting methods offer several advantages, primarily providing prompts to models that serve as instructions, guiding them on what specific task needs to be performed and how to approach it. Additionally, shot prompting methods supply a few examples to the models, aiding them in inference and comprehension for the underlying task. This approach enables the models to better understand the task and generate more

Table 4: present the characteristics of experimental datasets. Here, #sample indicates the number of news headlines and body pairs in the dataset. Similarly, #head and #Word indicate the average number of words in the headline and news body. Whereas #sent indicates the average number of sentences in the news body and #num indicates the average number of numeric figures in the news body.

	#sample	#head	#sent	#Word	#num
Train	21157	7.769	9.851	179.116	9.884
Dev	2367	7.723	9.719	178.396	9.595
Test	5227	8.082	10.427	190.006	10.186

accurate and contextually relevant headlines containing numeral figures. We consider GPT, $T5$ and $LLaMA$ in few shot prompt settings. From Table 1 it is apparent that $LLaMA$ the model outperforms GPT and $T5$ with few shot prompting. Similarly, it is also evident that $LLaMA$ a model with few shot prompting outperforms $RoBERTa$, GPT, $BART$, and $T5$ models in *Pretrained* and *Fine-tuned* setups. Our manual inspection of the news headlines generated by the GPT, $T5$, and $LLaMA$ models utilizing few-shot prompting over a subset of the dataset containing fifty samples suggests that the implementation of few-shot prompting enhances the efficiency of numeral-aware headline generation by the models. Based on the findings presented in Table 1, it's clear that few-shot prompting using the $LLaMA$ model outperforms both few-shot prompting with $T5$ and GPT. As a result, we chose to submit headlines generated by the few-shot prompting with the $LLaMA$ model as our final system for evaluation at NumEval-2024. We could have fine-tuned the $LLaMA$ model for better results, but we have only used the pre-trained $LLaMA$ model due to resource constraints.

Table 2 presents the human evaluation of headlines generated by our proposed system (few-shot

prompting with $LLaMA$), which were submitted to NumEval-2024. The organizers of NumEval-2024 conducted this human evaluation of the generated headlines. It is apparent from Table 2 that our proposed system (few-shot prompting with $LLaMA$) achieved the top rank in recommending 100 news.

5 Error Analysis

This study also conducts an error analysis to examine the strengths and weaknesses of large language models (LLMs) across different setups for numeral-aware headline generation. Through this analysis, we aim to identify patterns of errors and limitations inherent in the models, providing valuable insights into areas for improvement and optimization in future model development and training methodologies. We selected fifty news headline-body pairs, where numeral figures in the headline are crucial for accurately representing the news content. Our examination of the generated headlines by the models revealed the following insights: (i) $RoBERTa$ the model generates a headline, which is representative of the news body, however in some instances is failed to consider the numeric value for headline generation. Consequently, $RoBERTa$ is deemed unsuitable for numeral-aware headline generation. However, fine-tuning the $RoBERTa$ model enhances generated headline quality, which is also evident by the performance comparison between its *Pretrained* and *Fine-tuned* setups. (ii) The $BART$ models, whether in the *Pretrained* or *Fine-tuned* setups, demonstrate proficiency in generating efficient headlines that include valid numeric values. However, it is noteworthy that the inclusion of valid numeric values in headlines is more prevalent in the fine-tuned models compared to those without fine-tuning. (iii) The $T5$ models, in both the *Pretrained* and *Fine-tuned* setups, consistently produced headlines with more efficient and valid numerical values compared to $RoBERTa$, $BART$, and GPT. This indicates that $T5$ models are particularly more effective in numeral aware headline generations. (iv) The $LLaMA$ model stands out for its ability to generate accurate and efficient headlines containing valid numerical values when compared to $RoBERTa$, $BART$, GPT, and $T5$. This suggests that the $LLaMA$ model excels in incorporating precise numeric information into its generated headlines, surpassing other models in this aspect. Figure 2 presents an example' comparison between headline generated by fine-tuned $T5$ model and headline generated by $T5$ with three shot prompt. From Figure 2, it is apparent that the headline generated by $T5$ with three three-shot prompts better represents the news body compared to the headline generated by the fine-tuned T5 model. This further validates our claim that a few-shot prompt helps the LLMs generate headlines.

6 Conclusion and Future work

This study conducted an empirical research on LLMs for numeral-aware headline generation and proposed a few shots prompting with LLMs for numeral-aware headline generation. We conduct our experiments on *NumHG* and test set data provided by the organizer of NumEval-2024. Our experimental results suggest that finetuning LLMs over *NumHG* dataset improves the performance of numeral-aware headline generation. Further, few shot prompting with LLMs outperform fine-tuned LLMs for numeral-aware headline generation. This study identifies prompt tuning using LLMs for numeral-aware headline generation.

References

Lochan Basyal and Mihir Sanghvi. 2023. Text summarization using large language models: A comparative study of mpt-7b-instruct, falcon-7b-instruct, and openai chat-gpt models. *arXiv preprint arXiv:2310.10449*.

Wang Chen, Hou Pong Chan, Piji Li, and Irwin King. 2020. Exclusive hierarchical decoding for deep keyphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1095–1105, Online. Association for Computational Linguistics.

Sophie Chesney, Maria Liakata, Massimo Poesio, and Matthew Purver. 2017. Incongruent headlines: Yet another way to mislead your readers. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 56–61, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zijian Ding, Alison Smith-Renner, Wenjuan Zhang, Joel Tetreault, and Alejandro Jaimes. 2023. Harnessing the power of llms: Evaluating human-ai text co-creation through the lens of news headline generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3321–3339.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Maksym Gabielkov, Arthi Ramachandran, Augustin Chaintreau, and Arnaud Legout. 2016. Social clicks: What and who gets read on twitter? In *Proceedings of the 2016 ACM SIGMETRICS international conference on measurement and modeling of computer science*, pages 179–192.

Jian-Tao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2023. Numhg: A dataset for number-focused headline generation. *arXiv preprint arXiv:2309.01455*.

Sujit Kumar, Durgesh Kumar, and Sanasam Ranbir Singh. 2023. Gated recursive and sequential deep hierarchical encoding for detecting incongruent news articles. *IEEE Transactions on Computational Social Systems*.

Sujit Kumar, Gaurav Kumar, and Sanasam Ranbir Singh. 2022. Detecting incongruent news articles using multi-head attention dual summarization. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 967–977.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yutao Luo, Menghua Lu, Gongshen Liu, and Shilin Wang. 2022. Few-shot table-to-text generation with prefix-controlled generator. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6493–6504.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Timo Schick and Hinrich Schütze. 2021. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402.

Shengli Song, Haitao Huang, and Tongxiao Ruan. 2019. Abstractive text summarization using lstm-cnn based deep learning. *Multimedia Tools and Applications*, 78:857–875.

Percy H Tannenbaum. 1953. The effect of headlines on the interpretation of news stories. *Journalism Quarterly*, 30(2):189–197.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Daphne van Zandvoort, Laura Wiersema, Tom Huibers, Sandra van Dulmen, and Sjaak Brinkkemper. 2023. Enhancing summarization performance through transformer-based prompt engineering in automated medical reporting. *arXiv preprint arXiv:2311.13274*.

Wei Wei and Xiaojun Wan. 2017. Learning to identify ambiguous and misleading news headlines. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4172–4178.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

HierarchyEverywhere at SemEval-2024 Task 4: Detection of Persuasion Techniques in Memes Using Hierarchical Text Classifier

Omid Ghahroodi
NLP & DH Lab,
Computer Engineering Department
Sharif University of Technology, Tehran, IR
oghahroodi98@gmail.com

Ehsaneddin Asgari
Language Technologies Group
Qatar Computing Research Institute
Doha, Qatar
easgari@hbku.edu.qa

Abstract

Text classification is an important task in natural language processing. **H**ierarchical **T**ext **C**lassification (**HTC**) is a subset of text classification task-type. HTC tackles multi-label classification challenges by leveraging tree structures that delineate relationships between classes, thereby striving to enhance classification accuracy through the utilization of inter-class relationships. Memes, as prevalent vehicles of modern communication within social networks, hold immense potential as instruments for propagandistic dissemination due to their profound impact on users. In SemEval-2024 Task 4, the identification of propaganda and its various forms in memes is explored through two sub-tasks: (i) utilizing only the textual component of memes, and (ii) incorporating both textual and pictorial elements. In this study, we address the proposed problem through the lens of HTC, using state-of-the-art hierarchical text classification methodologies to detect propaganda in memes. Our system achieved first place in **English Sub-task 2a**, underscoring its efficacy in tackling the complexities inherent in propaganda detection within the meme landscape.

1 Introduction

1.1 Propaganda Techniques in Memes

Propaganda can be defined as the deliberate dissemination of information, often with a biased or misleading nature, aimed at promoting or publicizing a particular political cause, ideology, or viewpoint. This communication tactic takes various forms, including persuasive messaging, advertising campaigns, and the dissemination of ideas through media channels. The primary objective of propaganda is to influence people's beliefs, attitudes, or behaviors towards a specific agenda or ideology. Examples of propaganda can range from political advertisements designed to sway voters, to ideological messaging spread through social media platforms.

Memes have emerged as one of the most prevalent communication tools in digital media. Their utilization of both text and image allows for the transmission of substantial information, underscoring the critical need for detecting propaganda within them.

1.2 Task Overview

SemEval-2024 Task 4 (Dimitrov et al., 2024) addressed the challenge of propaganda technique detection within memes in three sub-tasks (**1**, **2a**, **2b**) and four languages (**English**, **Bulgarian**, **North Macedonian**, **Arabic**). The organizers focused on different aspects of meme analysis: **Task 1** concentrated on detecting propaganda techniques from the textual content of memes, while **Tasks 2a** and **2b** respectively tackled the identification of techniques and the presence or absence of propaganda in a multimodal format. The SemEval-2024 Task 4 introduced three distinct sub-tasks across four languages. **English** language data was provided in supervised learning, whereas **Bulgarian**, **North Macedonian**, and **Arabic** language datasets were presented in a zero-shot learning framework. It is important to note that this task presented propaganda techniques in the form of a hierarchy, illustrated in Figure 1.

1.3 Hierarchical Text Classification

Hierarchical **T**ext **C**lassification (**HTC**) is a method wherein classes are organized in a hierarchical structure. This approach aims to enhance the accuracy of text classification models by leveraging the relationships within this hierarchy. We used the previous state-of-the-art (SOTA) hierarchical text classification model (**HPT** (Wang et al., 2022b)) to identify propaganda techniques in memes based on the hierarchical structure of propaganda techniques.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1727–1732

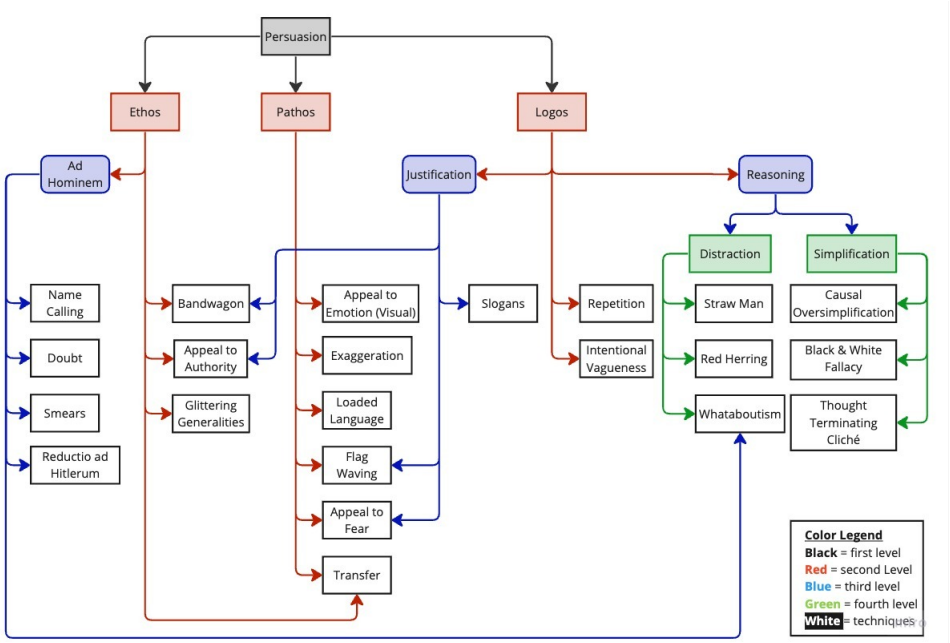


Figure 1: The diagram depicts propaganda techniques, represented as white nodes, organized in a directed acyclic graph (DAG). This image is sourced from the task description paper (Dimitrov et al., 2024)

In the multimodal section, we focused only on the textual content of memes, disregarding the accompanying images.

1.4 Our Discoveries

Our investigation revealed that employing hierarchical text classification models significantly enhanced text classification accuracy compared to various other methodologies and baseline approaches. Intriguingly, our decision to exclude the image component from consideration in **Task 2** resulted in the highest accuracy among all participating teams in **Task 2a**. We attribute this outcome to the inherent limitations of multimodal models in comprehending the intricate semantic relationships between images and text, particularly in the context of propaganda detection. Incorporating the image data would likely have increased model complexity and reduced accuracy, as observed in the performance of other teams. For the multilingual part, we rely on translation for non-English memes.

We utilized the HPT (Wang et al., 2022b) model source code[1], making necessary modifications to adapt it to our specific use case. The final version of our system's code has been made publicly available on GitHub for transparency and reproducibility[2].

[1]https://github.com/wzh9969/HPT

[2]https://github.com/language-ml/SemEval-2024-Task-4

2 Background

2.1 Dataset

The dataset utilized in this study comprises both textual and pictorial content extracted from memes along with associated propaganda technique tags (Dimitrov et al., 2024). Specifically, **Task 1** involves texts extracted from memes alongside propaganda technique tags, except **Loaded Language** and **Name Calling/Labeling** techniques, which are not included in the tags. **Task 2a** expands upon **Task 1** by incorporating images of memes, thereby presenting a multi-label classification task in a multi-modal format. **Task 2b** is similar to **Task 1a**, except that it involves binary classification regarding the presence or absence of propaganda. The organizers released the three tasks for the English language in a supervised manner and for **English**, **Bulgarian**, **North Macedonian**, and **Arabic** language in a zero-shot manner. The organizers released the dataset in three parts: training, validation, and testing sets.

2.2 Propaganda Detection

In recent research, the task of detecting propaganda in various forms of media has gained significant attention. (Da San Martino et al., 2020) introduce a task focused on identifying propaganda in news articles, comprising two subtasks: detecting spans containing propaganda and identifying

specific propaganda techniques from a predefined set of 14 techniques. On the other hand, (Dimitrov et al., 2021) presents a task aimed at detecting propaganda techniques in memes, without considering the hierarchy relation between techniques.

2.3 Hierarchical Text Classification

In this paper, we categorize existing hierarchical text classification models into three main categories: local methods, global methods, and generative methods.

1. **Local Methods**: Local methods tackle the hierarchical classification problem by addressing individual categories within the hierarchy. (Banerjee et al., 2019) employ binary classifications for each category and mitigate the issue of data scarcity at lower levels through transfer learning from parent to child categories. (Kowsari et al., 2017) adopt a strategy of training a multi-label classifier for each node, while (Dumais and Chen, 2000) employ SVM per level. (Shimura et al., 2018) leverage multiple CNNs to address classification at each level of the hierarchy.

2. **Global Methods**: Global methods take a holistic approach by employing a single classifier to predict all classes within the hierarchical structure. HiAGM (Zhou et al., 2020) utilizes two encoders, TreeLSTM and GCN, to derive the tree representation. They introduce two models, HiAGM-LA and HiAGM-TP, which respectively utilize attention mechanisms on classes and text propagation within the graph encoder. (Deng et al., 2021) aims to enhance the HiAGM by using information theory. (Mao et al., 2019) frame the hierarchical classification as a reinforcement learning problem, seeking an optimal policy for traversing suitable labels within the tree. (Zhu et al., 2023) employ structural entropy to construct the code tree, followed by using HiAGM-TP. (Wang et al., 2022a) introduce contrastive learning and positive samples to incorporate hierarchy into the text encoder. (Chen et al., 2021) attempt to unify label embedding and text embedding in a single space using triplet loss. In (Wang et al., 2022b), soft prompt tuning is employed, whereby each row of the hierarchy is fed into a graph attention network. Subsequently, the representations obtained from each row are provided as input to the BERT model. The model is trained to predict the correct label corresponding to the output of these tokens.

3. **Generative Methods**: (Yu et al., 2022) addressed the challenge of hierarchical classification by employing a method that generates a sequence of labels. Their approach involves training a T5 model to generate paths within the hierarchy. (Kwon et al., 2023) also tackled hierarchical classification through label generation. Notably, their approach enabled the model to generate n-grams not explicitly present in the predefined problem categories.

3 System overview

In **sub-task 1**, we addressed the proposed problem using hierarchical text classification and used a state-of-the-art (SOTA) HTC model for propaganda technique detection with some modifications. We utilized HPT (Wang et al., 2022b) as the hierarchical text classifier.

Convert Task to HTC Problem: The hierarchical structure of propaganda techniques was represented as a **D**irected **A**cyclic **G**raph (DAG). The hierarchy of propaganda techniques is depicted in Figure 1. To use the HPT model (Wang et al., 2022b), it was imperative to transform this DAG into a hierarchical tree. This transformation involved converting nodes with multiple parents into new nodes. For instance, the node "Whataboutism" with two parents, "Distraction" and "Ad hominem" was split into two nodes labeled "Distraction_Whataboutism" and "Ad hominem_Whataboutism". Two methods can be employed for organizing the first level of the hierarchy tree: **(1)** placing two nodes labeled "propagandistic" and "non-propagandistic" at the initial level, followed by the entire hierarchy of propaganda techniques under the "propagandistic" node, or **(2)** directly utilizing the hierarchy tree without this initial categorization. Our observations indicate that **method 1** yields superior performance.

Additional Datasets: We utilized two additional datasets, (Da San Martino et al., 2020) and (Dimitrov et al., 2021), as supplementary sources for training our model. In employing the data from (Da San Martino et al., 2020), we focused on its **TC sub-task**, which involves identifying propaganda techniques within news articles. The format of the data provided by (Da San Martino et al.,

Task	Model	HF1	HP	HR	Rank
English - Subtask 1	Best model	0.75247	0.68419	0.83590	1/33
	Our system	0.64252	0.63618	0.64899	12/33
	Our system†	0.65286†	0.63041†	0.67697†	9/34†
	Baseline	0.36865	0.47711	0.30036	31/33
English - Subtask 2a	Our system	0.74592	0.86682	0.65461	1/14
	Baseline	0.44706	0.68778	0.33116	13/14
Bulgarian - Subtask 1	Best model	0.56833	0.51955	0.62722	1/20
	Our system	0.46757	0.48301	0.45310	9/20
	Baseline	0.28377	0.31881	0.25567	18/20
Bulgarian - Subtask 2a	Best model	0.62693	0.70278	0.56586	1/8
	Our system	0.46414	0.67080	0.35483	7/8
	Baseline	0.50000	0.80428	0.36276	5/8
North Macedonian - Subtask 1	Best model	0.51244	0.51824	0.50677	1/20
	Our system	0.41713	0.48609	0.36531	10/20
	Baseline	0.30692	0.31403	0.30012	17/20
North Macedonian - Subtask 2a	Best model	0.63681	0.75019	0.55320	1/8
	Our system	0.35693	0.68903	0.24085	8/8
	Baseline	0.55525	0.90219	0.40103	4/8
Arabic - Subtask 1	Best model	0.47593	0.39140	0.60702	1/17
	Our system	0.40545	0.35638	0.47018	7/17
	Baseline	0.35897	0.35000	0.36842	14/17
Arabic - Subtask 2a	Best model	0.52613	0.55311	0.50166	1/8
	Our system	0.43685	0.50998	0.38206	6/8
	Baseline	0.48649	0.65000	0.38870	3/8

Table 1: The table presents the performance results of the hierarchical text classification model in comparison to both the baseline model and the best-performing model in sub-tasks 1 and 2a across four different languages: English, Bulgarian, North Macedonian, and Arabic. For each sub-task, the metrics HF1 (hierarchical F1 score), HP (hierarchical precision), and HR (hierarchical recall) are reported. † refers to the model trained initially on the (Dimitrov et al., 2021) dataset and subsequently fine-tuned on the task dataset, submitted after the test phase.

Task	Model	F1 macro	F1 micro	Rank
English - Subtask 2b	Best model	0.81030	0.82500	1/20
	Our system	0.56309	0.66167	16/20
	Baseline	0.25000	0.33333	20/20
Bulgarian - Subtask 2b	Best model	0.67100	0.81000	1/15
	Our system	0.48547	0.63000	10/15
	Baseline	0.16667	0.20000	15/15
North Macedonian - Subtask 2b	Best model	0.68627	0.84000	1/15
	Our system	0.50624	0.62000	6/15
	Baseline	0.09091	0.10000	15/15
Arabic - Subtask 2b	Best model	0.61487	0.63125	1/15
	Our system	0.56196	0.66875	5/15
	Baseline	0.22705	0.29375	15/15

Table 2: The table presents the performance results of the hierarchical text classification model in comparison to both the baseline model and the best-performing model in sub-task 2b across four different languages: English, Bulgarian, North Macedonian, and Arabic. For each sub-task, the Macro F1 and Micro F1 are reported.

2020) consists of spans within the news text annotated with corresponding propaganda techniques. To integrate this data into our model, we adopted an approach where if a span within a news article contained a propaganda technique, we assigned that particular technique to the entire article. It's important to note, however, that the dataset from (Da San Martino et al., 2020) does not encompass all the propaganda techniques featured in the SemEval-2024 task 4 dataset. Our analysis revealed that utilizing the data from (Da San Martino et al., 2020) in this manner led to a decrease in model accuracy. We attribute this reduction to two primary factors: **(1)** the broad attribution of propaganda techniques to entire news articles and **(2)** the differing distribution characteristics between news articles and meme text. The task of detecting propaganda techniques from memes, as outlined in (Dimitrov et al., 2021), served as another additional dataset for our study. Our analysis revealed that incorporating the data provided by (Dimitrov et al., 2021) enhanced the accuracy of our model.

[CLS] Token: Many memes comprise multiple sentences distributed across different picture boxes, delineated by "\n\n" in the dataset. To establish coherence between sentence boundaries, we utilized **"[CLS]" Token** between sentences. We observed that the inclusion of this token between sentences improves the performance of the model.

Other Tasks: In subtasks **2a** and **2b**, the image component of memes was disregarded, and only the textual content was provided to the model. Furthermore, for all the sub-tasks that are non-English, we used Google Translation API to translate them into English and used the model of the previous part

Baseline: According to the task description, the baseline for each sub-task is the most common label.

4 Experimental Setup

The organizers provided the data in three parts: training, evaluation, and testing sets. We employed the HPT model, utilizing the **bert-base-uncased** language model for our study. For training purposes, we combine the training and evaluation data, randomly picking **10%** for evaluation, and reserving the remaining **90%** for training. Our training comprised a batch size of **8** and a learning rate of **3e-5**. The remaining hyperparameters are similar to the HPT paper. To use additional data, we initially trained the model on this additional dataset before continuing training on the task data.

5 Results

The results for **sub-tasks 1** and **2a** are presented in Table 1, while the outcomes for **sub-task 2b** are shown in Table 2. Our system has exhibited strong performance in English language Task 1. In **sub-task 2a** for English, despite our model solely leveraging textual content from memes without considering images, it achieved the top ranking. We attribute this observation to two main factors: **(1)** The challenge of discerning the connection between images and text in propaganda detection **(2)** A substantial portion of the requisite information for propaganda detection likely resides within the textual component in addition to the image itself.

However, our system encountered challenges in non-English sub-tasks, displaying poor performance. We attribute this to potential translation errors and the absence of a pre-processing pipeline for these languages.

6 Conclusion

In this study, we addressed the challenge of detecting propaganda techniques in memes through two distinct sub-tasks: textual and multimodal analysis, conducted in both supervised and zero-shot settings across various languages. To tackle this issue, we employed hierarchical text classification. In the multimodal sub-tasks, we focused solely on the textual content of memes, achieving notable performance. However, when dealing with sub-tasks in languages other than English, our system's performance suffered. We concluded by presenting the metrics and conducting a thorough analysis of the results. Moving forward, our next objective is to develop a better hierarchical text classification model with better performance.

References

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. 2019. Hierarchical transfer learning for multi-label text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300, Florence, Italy. Association for Computational Linguistics.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021. Hierarchy-aware label semantics matching

network for hierarchical text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4370–4379, Online. Association for Computational Linguistics.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Zhongfen Deng, Hao Peng, Dongxiao He, Jianxin Li, and Philip Yu. 2021. HTCInfoMax: A global model for hierarchical text classification via information maximization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3259–3265, Online. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Susan Dumais and Hao Chen. 2000. Hierarchical classification of web content. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 256263, New York, NY, USA. Association for Computing Machinery.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 364–371.

Jingun Kwon, Hidetaka Kamigaito, Young-In Song, and Manabu Okumura. 2023. Hierarchical label generation for text classification. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 625–632, Dubrovnik, Croatia. Association for Computational Linguistics.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. Hierarchical text classification with reinforced label assignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 445–455, Hong Kong, China. Association for Computational Linguistics.

Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2018. HFT-CNN: Learning hierarchical category structure for multi-label short text categorization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 811–816, Brussels, Belgium. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022a. Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7109–7119, Dublin, Ireland. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022b. HPT: Hierarchy-aware prompt tuning for hierarchical text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3740–3751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chao Yu, Yi Shen, and Yue Mao. 2022. Constrained sequence-to-tree generation for hierarchical text classification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 18651869, New York, NY, USA. Association for Computing Machinery.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117, Online. Association for Computational Linguistics.

He Zhu, Chong Zhang, Junjie Huang, Junran Wu, and Ke Xu. 2023. HiTIN: Hierarchy-aware tree isomorphism network for hierarchical text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7809–7821, Toronto, Canada. Association for Computational Linguistics.

AILS-NTUA at SemEval-2024 Task 9: Cracking Brain Teasers: Transformer Models for Lateral Thinking Puzzles

Ioannis Panagiotopoulos, Giorgos Filandrianos, Maria Lymperaiou, Giorgos Stamou
School of Electrical and Computer Engineering, AILS Laboratory
National Technical University of Athens
yiannispn@gmail.com, {geofila, marialymp}@islab.ntua.gr, gstam@cs.ntua.gr

Abstract

In this paper, we outline our submission for the SemEval-2024 Task 9 competition: 'BRAIN-TEASER: A Novel Task Defying Common Sense'. We engage in both sub-tasks: Sub-task A-Sentence Puzzle and Sub-task B-Word Puzzle. We evaluate a plethora of pre-trained transformer-based language models of different sizes through fine-tuning. Subsequently, we undertake an analysis of their scores and responses to aid future researchers in understanding and utilizing these models effectively. Our top-performing approaches secured competitive positions on the competition leaderboard across both sub-tasks. In the evaluation phase, our best submission attained an average accuracy score of 81.7% in the Sentence Puzzle, and 85.4% in the Word Puzzle, significantly outperforming the best neural baseline (ChatGPT) by more than 20% and 30% respectively.

1 Introduction

In Natural Language Processing (NLP), reasoning serves as the cognitive backbone, enabling systems to transcend mere language comprehension and delve into sophisticated understanding. Despite the excellence of Large Language Models (LLMs) in several linguistic tasks, their reasoning capabilities are still questionable to a non-negligible extend (Floridi and Chiriatti, 2020; Bender et al., 2021; Kauf et al., 2022; Zhang et al., 2023; Shi et al., 2023; Tyen et al., 2024; Giadikiaroglou et al., 2024), often posing the fundamental concerns of whether they can indeed reason or memorize exhaustively (Yuan et al., 2022).

Such limitations can be probed via well-crafted datasets and benchmarks, showcasing varying LLM deficiencies at a time. As the core of the current paper, BrainTeaser (Jiang et al., 2023b, 2024b) incorporates problems that stress models to think "out-of-the-box"; to this end, the key novelty of BrainTeaser is that in order to answer correctly, models need to defy default senses of concepts and common associations. Surprisingly, state-of-the-art (SoTa) LLMs, such as ChatGPT can only exhibit a maximum accuracy of ∼60% when solving BrainTeaser riddles, demonstrating an inherently limited reasoning ability in unconventional thinking.

Thus, assuming that large-scale training and prompting may not always serve as universally applicable solutions towards flexible reasoning, we move one step back and leverage transfer learning techniques starting from smaller models based on masked language modelling, such as BERT (Devlin et al., 2019) and consequent BERT-based encoders. Then, we proceed with similar techniques on LLMs, aiming to showcase that significant performance advancements using a small set of in-domain data for parameter updating can be achieved in comparison to merely querying the model's prior knowledge via prompting. Therefore, our contributions are:

1. We perform lightweight tuning on smaller encoder models and LLMs, significantly outperforming the reported baselines.
2. We transform the multiple-choice problem to a binary classification one, aiming to explore diverging reasoning paths for models.
3. We ground final performance on the models' "prior knowledge" in related problems.
4. We delve into models' frequent failures to obtain a deeper understanding of reasoning cues that make models struggle the most.

Our code is available on GitHub [1].

2 Related work

Reasoning in NLP has enjoyed several advancements due to the surge of pre-trained language mod-

[1] https://github.com/GiannisPana/AILS-NTUA-at-SemEval-2024-Task-9-Brainteaser

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1733–1746

els and especially LLMs (Sun et al., 2023). Reasoning challenges incorporate commonsense reasoning (Richardson and Heck, 2023), involving inference regarding everyday situations, mathematical reasoning (Lu et al., 2023), referring to the ability of solving mathematical problems, logical reasoning (Yang et al., 2023), which includes the systematic deduction of conclusions based on established principles and formal rules, causal reasoning (Gendron et al., 2024), which studies cause-and-effect relationships explaining why an event leads to another, and several other sub-tasks (Vashishtha et al., 2020; Wei et al., 2023; Petersen and van der Plas, 2023). In terms of reasoning evaluation, BigBench (Srivastava et al., 2023) comprises 204 reasoning tasks, targeting to explore the related capabilities of recent LLMs. Several dedicated datasets have been developed to tackle different reasoning challenges, including commonsenseQA (Talmor et al., 2019), WinoGrande (Sakaguchi et al., 2019), RiddleSense (Lin et al., 2021) and others; most of these datasets are incorporated in Tasksource (Sileo, 2023). Especially RiddleSense questions aspects of reasoning close to BrainTeaser (Jiang et al., 2023b, 2024b).

3 Task and Dataset Description

The BrainTeaser task at SemEval-2024 (Jiang et al., 2023b, 2024b) features lateral thinking puzzles presented as multiple-choice questions (QAs). Each question offers four options, with one being the correct answer and the others serving as distractors. Additionally, the final option is always "None of above". It consists of two sub-tasks, *Task A*: **Sentence Puzzle** and *Task B*: **Word Puzzle**. In addition to the original puzzles, the dataset includes adversarial subsets created by manually modifying the original brain teasers while preserving their reasoning paths. The original data were perturbed in two ways: First, there is *semantic reconstruction* of each original question without altering the answers or the distractors. Second, the original data underwent *context reconstruction*, wherein the original reasoning path remains intact, but the brain teaser describes a new situational context. Overall, the dataset used for training and evaluation consists of triplets of data: original, semantic, and context reconstruction. Table 1 provides an example of the triplets of data that constitute the dataset.

***Task A*: Sentence Puzzle** In this sub-task, the sentence pairs are crafted in a manner that makes it relatively easy for humans to discern the correct statement, yet challenging for systems, even those equipped with commonsense understanding. Table 2 contains examples of the Sentence Puzzle dataset (on the left). The training data consists of 169 distinct multiple-choice QA sets, each accompanied by its semantic and context reconstructions, resulting in a total of 507 multiple-choice questions (3×169).

Question	Choice
Original	
What kind of nut has no shell?	A peanut.
	A doughnut.
	A walnut.
	None of above.
Semantic Reconstruction	
Which nut doesn't have a shell?	**A doughnut.**
	A walnut.
	A peanut.
	None of above.
Context Reconstruction	
Which type of bell doesn't make a sound?	A fire bell.
	A cow bell.
	A bluebell.
	None of above.

Table 1: Illustration of the structure of each sub-task's dataset, showcasing the original statement along with its two adversarials.

***Task B*: Word Puzzle** involves word-type brain teasers, where the answer defies the default meaning of the word and focuses on the letter composition of the question. The training dataset comprises 132 multiple-choice QAs, each accompanied by its semantic and context reconstructions, resulting in a total of 396 multiple-choice QAs (3×132). These brain teaser categories include puns, homophones, ambiguous words, and various other linguistic puzzles, as showcased in the examples provided in Table 2 on the right-hand side. The Word Puzzle sub-task pose challenges not only for systems but also for humans in discerning the correct answer.

Data statistics The BrainTeaser dataset comprises 3 data splits, namely train, development (used during the practice phase), and the hidden test set, which was used for evaluation. Statisics are provided in Table 3. Throughout the evaluation phase, the leaderboard was kept concealed.

Evaluation Metrics Both sub-tasks are assessed via accuracy metrics to gauge the performance of participating systems in two ways. First, instance-based accuracy evaluates each question individually, considering original questions and their seman-

Sentence Puzzle		*Word Puzzle*	
Question	**Choice**	**Question**	**Choice**
A man shaves everyday, yet keeps his beard long.	**He is a barber.**	What has toes but no feet or legs?	Cabbages.
	He wants to maintain his appearance.		**Tomatoes.**
	He wants his girlfriend to buy him a razor.		Onions.
	None of above.		None of above.
You go to the doctor because you're sick, and he gives you three medicines to take every half hour. How long do the drugs keep you going?	One and a half hours.	What did the little lobster get on its math test?	**Sea-plus.**
	Two hours.		Very-bad.
	An hour.		Very-Good.
	None of above.		None of above.
How many times can you deduct 10 from 100?	**Once.**	What's the beginning of an argument?	The letter T.
	Infinite time.		**The letter A.**
	Twice.		The letter U.
	None of above.		None of above.

Table 2: Example questions illustrating both sub-tasks, with correct answers highlighted in bold. Examples on the left pertain to *sub-task A: Sentence Puzzle*, while those on the right correspond to *sub-task B: Word Puzzle*.

Sub-task	**Train**	**Dev**	**Test**
A - Sentence Puzzle	507	120	120
B - Word Puzzle	396	96	96

Table 3: Data statistics.

tic and context adversarials. This metric provides a detailed understanding of a model's proficiency in reasoning through various scenarios. In contrast, group-based accuracy takes a broader perspective, assessing questions and associated adversarials as cohesive groups. Each group consists of three questions, and a model scores 1 only if it correctly solves *all* questions in a group. This approach evaluates the system's holistic performance in navigating through lateral thinking challenges. The combined use of instance-based and group-based accuracy metrics provides comprehensive insights into the capabilities of participating systems in tackling the complexities of both sub-tasks.

4 Methods

We focus on tuning language models belonging into two categories. First, we fine-tune variations of *encoder* models, namely BERT (Devlin et al., 2019), RoBERTa-large (Liu et al., 2019) and DeBERTaV3-base (He et al., 2023), to assess the impact of transfer learning using various datasets requiring similar reasoning abilities, apart from BrainTeaser. We study the problem using the provided *multi-choice* setup, but we also transform it into a *binary* classification task. Secondly, the encoders' results are compared with those obtained from *fine-tuned LLMs* using the BrainTeaser dataset. To achieve this, we fine-tune Llama 2 (Touvron et al., 2023b), Phi-2 (Gunasekar et al., 2023) and Mistral-7b (Jiang et al., 2024a), which have already demonstrated enhanced reasoning abilities. In this regard, we examine the effect of the model size on our task, which has already been reported in the literature to significantly influence the reasoning abilities of the models (Touvron et al., 2023b; Wei et al., 2022), along with other tuning hyperparameters. Model details are presented in App. A.

4.1 Encoder models

Pre-training First, we evaluate the effects of the pre-training on our task. Thus, we select two variations of each encoder: the *vanilla* one (using the default pre-trained basis and fine-tuned on BrainTeaser data only) and one that has undergone additional pre-training using supplementary commonsense reasoning datasets before fine-tuned on BrainTeaser. In the second case, we use the following pre-trained models: (1) BERT-SE: a BERT-base-uncased version pre-trained on the multiple-choice dataset used in SemEval-2020 Task 4b (Wang et al., 2020) (2) RoBERTa-WNGRD: a RoBERTa-large version pre-trained on the WinoGrande dataset, and (3) DeBERTaV3-TS: a DeBERTaV3-base model, pre-trained on diverse commonsense reasoning datasets, and fine-tuned with multi-task learning on over 600 tasks from the Tasksource collection.

Multi-class Classification task This strategy involves treating the problem as multi-class classification: all four provided options are combined with the given question, and consequently these concatenated inputs are fed into the model, which is fine-tuned to select one of the four options as part of a multi-class classification problem.

Binary Classification task Each sample originally consisting of multiple-choice QAs with four available options, underwent the following transformation: each candidate answer (excluding the

"None of above" option) was paired with the question receiving the label 0 if the choice was incorrect, or the label 1 for the opposite. In case all the 3 pairings returned 0, it is directly implied that "None of above" is the correct answer.

4.2 LLMs

We demonstrate an in-depth examination of fine-tuning SoTa LLMs (Llama 2, Phi-2, and Mistral-7b) in the context of multi-class classification. Note that during inference, the models prompted to provide an *explanation* along with the label. This experimental step, which we have observed to improve the performance of the model, also provides a qualitative identification of flaws in the models' reasoning process. In our experiments, we explore various combinations of LoRA (Hu et al., 2021) a and r hyperparameters, using values of 16, 32, 64, and 128. For the analysis ahead, LLMs are denoted as model_r_a, reflecting these hyperparameters. Additional technical information, including prompting details and specifics about QLoRA hyperparameters, is available in App. B, C, D.

5 Experimental Results

Our metrics for the Sentence Puzzle sub-task are presented in Table 4 and for the Word Puzzle sub-task in Table 5 along with their baselines. Interestingly, the performance of the binary classification problem is significantly lower than that of the multi-class classification task. Initially, this behavior seemed counterintuitive since it appeared easier to determine whether a question is correct or not than to select the correct answer from four different options. However, this assumption is not accurate. Consider the word riddle: *'What is the capital in France?"* At first glance, the option 'F' seems incorrect, but when considering the options 'F,' 'E', 'A', and 'None of the above', 'F' emerges as the only correct answer, as it becomes apparent that the question refers to the capital *letter* rather than the capital *city*. Therefore, the diverse options provide crucial context to the models, explaining the superior performance of multi-class models. This lack of context is why we refrain from further exploring this methodology across all models in our study.

***Task A*: Sentence Puzzle** Table 4 illustrates minimal fluctuations among all instance-based metrics. This consistency extends to the associated group-based metrics for all models, highlighting a systematic behavior towards detecting various reasoning paths. This observation holds for both the encoder-based classifiers and LLMs utilized in this sub-task. Sentence puzzles inherently offer more detailed information, enabling models to detect and identify the same reasoning patterns more readily, regardless of changes in context, in contrast to word puzzles, which typically feature shorter contextual statements, presenting a greater challenge for models to discern consistent reasoning patterns.

Initially, it becomes apparent that pre-training encoders across various commonsense reasoning datasets results in substantial performance enhancements, as it enables the system to grasp domain-agnostic features which prove advantageous for the subsequent task. Additionally, several commonsense pre-trained encoders fine-tuned on BrainTeaser data outperform Llama 2 and Phi-2.

Another noteworthy observation from Table 4 is that only Mistral-7b from LLMs is able to surpass the encoder-type networks, while both Llama 2 and Phi-2 consistently scored lower. Unlike Llama 2 and Mistral-7b, Phi-2 has not undergone instruction fine-tuning (Gunasekar et al., 2023), which, coupled with the limited number of examples in the BrainTeaser Sentence Puzzle dataset, contributes to its lower performance, as a result of Phi's incapability to capture the complexities of the BrainTeaser data. In this regard, Mistral-7b, which has already demonstrated superior performance compared to every Llama 2 variation when tested in commonsense reasoning benchmarks (Jiang et al., 2023a), is also capable of solving this task more accurately.

***Task B*: Word Puzzle** In Table 5, we observe a stark contrast in the models' performance in understanding and detecting reasoning paths when the context changes. There are notable discrepancies in accuracy between original and semantic contexts when compared to context reconstruction, particularly evident in the case of smaller encoder models.

Regarding encoders, it is evident that, especially vanilla RoBERTa-large lacks robust commonsense reasoning and struggles to systematically handle ambiguity; in contrast, RoBERTa-large pre-trained on WinoGrande presents competitive performance. This notable enhancement (over 40%) due to WinoGrande pre-training suggests that this particular dataset effectively equips the model with the ability to understand word puzzle-related reasoning complexities, making its scores competitive with DeBERTaV3 in this sub-task, despite the higher

System	Original	Semantic	Context	Ori. + Sem.	Ori. + Sem. + Con.	Overall
Multi-class classification problem						
Human	.907	.907	.944	.907	.889	.920
ChatGPT	.608	.593	.679	.507	.397	.627
RoBERTa-L	.435	.402	.464	.330	.201	.434
Mistral-7b_128_128	**.850**	**.825**	**.775**	**.825**	**.700**	**.817**
Mistral-7b_64_128	.850	.825	.775	.825	.700	.817
Mistral-7b_16_64	.800	.800	.850	.750	.725	.817
Mixtral-8x7b_128_128	.850	.825	.725	.800	.700	.800
Llama 2-7b_64_128	.725	.650	.700	.575	.475	.692
Llama 2-13b_64_64	.665	.614	.645	.550	.400	.641
Llama 2-7b_64_64	.625	.600	.675	.550	.400	.633
Llama 2-7b_64_32	.250	.250	.425	.075	.000	.308
Phi-2_64_128	.625	.575	.550	.525	.425	.583
Phi-2_128_128	.625	.575	.550	.500	.375	.583
Phi-2_64_64	.525	.425	.550	.375	.300	.500
RoBERTa-WNGRD	.800	.775	.775	.750	.675	.784
DeBERTaV3-TS	.800	.775	.725	.750	.625	.767
DeBERTaV3-base	.725	.750	.675	.725	.625	.717
BERT-SE	.750	.725	.650	.700	.550	.708
RoBERTa-large	.700	.700	.725	.675	.550	.708
BERT	.675	.650	.650	.600	.475	.658
Binary classification problem						
DeBERTaV3-TS	.725	.650	.550	.650	.650	.642
RoBERTa-WNGRD	.575	.600	.500	.550	.550	.558
BERT-SE	.625	.550	.375	.525	.525	.517

Table 4: Model Performance for *sub-task A: Sentence Puzzle*. More results in Table 7.

System	Original	Semantic	Context	Ori.+Sem.	Ori.+Sem.+Con.	Overall
Multi-class classification problem						
Human	.917	.917	.917	.917	.900	.917
ChatGPT	.561	.524	.518	.439	.292	.535
RoBERTa-L	.195	.195	.232	.146	.061	.207
Mistral-7b_16_64	**.875**	**.906**	**.781**	**.813**	**.719**	**.854**
Mistral-7b_128_128	.844	.844	.813	.719	.625	.833
Mistral-7b_8_16	.781	.938	.781	.719	.562	.833
Mixtral-8x7b_128_128	.625	.719	.625	.531	.375	.656
Llama 2-13b_64_64	.354	.344	.438	.125	.031	.379
Llama 2-7b_64_64	.375	.344	.375	.125	.031	.365
Llama 2-7b_64_128	.281	.188	.438	.031	.031	.302
Phi-2_64_64	.688	.625	.688	.562	.438	.667
Phi-2_64_128	.656	.656	.625	.594	.406	.646
Phi-2_16_64	.625	.500	.688	.438	.312	.604
DeBERTaV3-base	.750	.750	.562	.656	.438	.687
DeBERTaV3-TS	.812	.781	.406	.719	.281	.666
RoBERTa-WNGRD	.750	.656	.500	.625	.312	.635
BERT	.562	.594	.469	.562	.312	.542
BERT-SE	.562	.500	.406	.500	.281	.489
RoBERTa-large	.281	.344	.344	.156	.094	.323
Binary classification problem						
DeBERTaV3-TS	.250	.313	.313	.219	.219	.292
RoBERTa-WNGRD	.375	.313	.125	.281	.281	.271
BERT-SE	.000	.000	.000	.000	.000	.000

Table 5: Model performance for *sub-task B: Word Puzzle*. More results in Table 7.

DeBERTaV3-base performance over RoBERTa-large in baseline reasoning benchmarks (He et al., 2023). Other than that, pre-training on other commonsense reasoning datasets does not significantly improve the overall performance for encoders. Conclusively, apart from WinoGrande the rest of the extra pre-training datasets do not hold reasoning cues close to BrainTeaser's word puzzles.

Regarding LLMs, Mistral-7b notably outperformed all others by a significant margin, even surpassing the 8 times larger model tuned using the same hyperparameters (*Mixtral-8x7b*). Llama 2

Example of correct prediction with explanation	
Question	**Choice**
If a dog is tied to a piece of rope that is 6m long how can it reach a bone that is 7m away?	**The other end is not tied to anything.**
	The rope stretches proportionally, providing the extra length needed for the horse to reach the hay seven meters away.
	The dog dug a 1m tunnel underground to reach the bone that was 7m away while still tied to the 6m long rope.
	None of above.
Explanation: A dog cannot reach a bone that is 7m away if it is tied to a piece of rope that is 6m long. This is because the distance between the dog and the bone is greater than the length of the rope. Therefore, the dog will not be able to reach the bone unless the rope is untied.	
Example of misprediction with explanation	
Question	**Choice**
How can you drop a raw egg onto a concrete floor and not crack it?	You need to drop the egg gently.
	You need to drop the egg hard.
	No problem, because concrete floors are very hard to crack.
	None of above.
Predicted Answer: None of above. **Explanation:** There is no way to drop a raw egg onto a concrete floor and not crack it. Concrete is a very hard material and will easily break any object that falls on it. The only way to avoid this would be to use some sort of protective covering for the egg before dropping it.	

Table 6: An example of explanations generated by Mistral-7b, while addressing the multi-class classification problem. The explanations, regardless of their validity, are detailed and showcase a high level of comprehension and reasoning. Correct answers are indicated in bold, while mispredictions reference the predicted answer by the model.

exhibited the worst results regardless of size (7/13 billion) and LoRA hyperparameters (r and a). Conversely, Phi-2 demonstrated relatively better performance, particularly considering its smaller parameter count (2.7 billion) compared to the other LLMs. However, both models performed worse compared to most fine-tuned encoders. This observations strongly confirms that word puzzles possess a distribution that diverges from the analytical commonsense reasoning required for sentence puzzles, entailing a unique set of cognitive demands.

Mistral-7b exhibits a trend where higher quality explanations were generated with higher values of lora rank r. However, the top-performing model showcased a configuration with r=16 and a=64. The QLoRA method (Hu et al., 2021) explains why our top model has a rank of 16 instead of 128, contrary to common expectations (more details reagrding QLoRA hyperparameters in App. C). Drawing from the widespread presence of low-rank structures, as highlighted by prior studies (Li et al., 2016, 2019; Grasedyck et al., 2013), we leverage the intrinsic low-rank structure in our problem, as emphasized in Hu et al. (2021). It is well-established that many tasks, particularly involving heavily overparametrized models, exhibit low-rank properties post-training (Oymak et al., 2019).

Overall, our systems demonstrate remarkably high overall accuracy, being less than 10% lower than human performance and more than 30% greater than ChatGPT. This suggests our methods' proficiency in understanding and detecting wordplay patterns, consistently addressing ambiguity irrespective of contextual and semantic variations in brain teasers. Upon reviewing the short explanations provided with each prediction (Table 6), we note thorough justifications even for incorrect answers. Errors typically adhere to specific wordplay patterns across original, semantic, and context multiple-choice questions (details in App. E).

6 Conclusion

In this study, we systematically evaluate pre-trained and fine-tuned encoders, along with instruction-tuned Large Language Models (LLMs), against two multi-class classification sub-tasks within the "BRAINTEASER: A Novel Task Defying Common Sense". We achieve competitive performance in both sub-tasks, accompanied by a plethora of insights regarding the influence of leveraging in-domain data, the variability model scale and architecture introduce, as well as the examination of diverging reasoning paths. As future work, we will delve into further reasoning patterns LLMs tend to follow with regard to lateral thinking challenges.

References

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

L. Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694.

Gaël Gendron, Michael Witbrock, and Gillian Dobbie. 2024. A survey of methods, challenges and perspectives in causality.

Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. 2024. Puzzle solving using reasoning of large language models: A survey.

Lars Grasedyck, Daniel Kressner, and Christine Tobler. 2013. A literature survey of low-rank tensor approximation techniques.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024b. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023b. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Carina Kauf, Anna A. Ivanova, Giulia Rambelli, Emmanuele Chersoni, Jingyuan S. She, Zawad Chowdhury, Evelina Fedorenko, and Alessandro Lenci. 2022. Event knowledge in large language models: the gap between the impossible and the unlikely. *ArXiv*, abs/2212.01488.

Yuanzhi Li, Yingyu Liang, and Andrej Risteski. 2016. Recovery guarantee of weighted low-rank approximation via alternating minimization.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. 2019. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2023. A survey of deep learning for mathematical reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14605–14631, Toronto, Canada. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. 2019. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian.

Molly R. Petersen and Lonneke van der Plas. 2023. Can language models learn analogical reasoning? investigating training objectives and comparisons to human performance.

Christopher Richardson and Larry Heck. 2023. Commonsense reasoning for conversational ai: A survey of the state of the art.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Huai hsin Chi, Nathanael Scharli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. *ArXiv*, abs/2302.00093.

Damien Sileo. 2023. tasksource: Structured dataset preprocessing annotations for frictionless extreme multi-task learning and evaluation. *arXiv preprint arXiv:2301.05948*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Co-

hen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, Yue Wu, Wenhai Wang, Junsong Chen, Zhangyue Yin, Xiaozhe Ren, Jie Fu, Junxian He, Wu Yuan, Qi Liu, Xihui Liu, Yu Li, Hao Dong, Yu Cheng, Ming Zhang, Pheng Ann Heng, Jifeng Dai, Ping Luo, Jingdong Wang, Ji-Rong Wen, Xipeng Qiu, Yike Guo, Hui Xiong, Qun Liu, and Zhenguo Li. 2023. A survey of reasoning with foundation models.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Gladys Tyen, Hassan Mansoor, Victor Cărbune, Peter Chen, and Tony Mak. 2024. Llms cannot find reasoning errors, but can correct them!

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online. Association for Computational Linguistics.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan

Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of The 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jingxuan Wei, Cheng Tan, Zhangyang Gao, Linzhuang Sun, Siyuan Li, Bihui Yu, Ruifeng Guo, and Stan Z. Li. 2023. Enhancing human-like multi-modal reasoning: A new challenging dataset and comprehensive framework.

Zonglin Yang, Xinya Du, Rui Mao, Jinjie Ni, and Erik Cambria. 2023. Logical reasoning over natural language as knowledge representation: A survey.

Zhangdie Yuan, Songbo Hu, Ivan Vulic, Anna Korhonen, and Zaiqiao Meng. 2022. Can pretrained language models (yet) reason deductively? *ArXiv*, abs/2210.06442.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023. How language model hallucinations can snowball.

A Model Selection

A.1 Encoder

BERT (Devlin et al., 2019): Bidirectional Encoder Representations for Transformers, is a pretrained deep bidirectional transformer model producing context representations. Using a fine-tuning setting, BERT has advanced state-of-the-art performances on a wide range of NLP tasks.

RoBERTa-large (Liu et al., 2019): Robustly Optimized BERT pre-training Approach (RoBERTa) is an adaptation of BERT architecture trained with larger batches on 160 GB data from various domains. RoBERTa-large was trained by dynamically modifying language masking while the next sentence prediction loss used in BERT was dropped. Other improvising techniques like larger input text sequences, byte pair encoding are used in training which seemingly improved the model performance in downstream tasks.

DeBERTaV3 (He et al., 2023): Decoding-enhanced BERT with disentangled attention is an extension of the original DeBERTa model. It builds upon the BERT (Bidirectional Encoder Representations from Transformers) architecture, aiming to enhance its decoding capabilities and overall performance across various natural language processing (NLP) tasks. DeBERTaV3 further improves the efficiency of DeBERTa (He et al., 2021) using ELECTRA-Style pre-training with Gradient Disentangled Embedding Sharing. Compared to DeBERTa, V3 significantly improves the model performance on downstream tasks. It incorporates a disentangled attention mechanism to allow the model to focus on different aspects of input independently, improving its ability to capture diverse linguistic patterns. The model also features enhancements in the decoding process, enabling more accurate text generation and sequence classification.

A.2 LLMs

Mistral-7b (Jiang et al., 2023a): Developed by EleutherAI, is a language model tailored for large-scale natural language processing tasks. With its 7 billion parameters, it excels in handling complex language understanding and generation tasks. Designed to perform exceptionally well across various NLP applications such as text generation, comprehension, and summarization, Mistral-7b surpasses the best open 13b model, Llama 2 (Touvron et al., 2023b), and the best released 34b model, Llama 1 (Touvron et al., 2023a), in reasoning, mathematics, and code generation tasks. Leveraging grouped-query attention (GQA) and sliding window attention (SWA), Mistral-7b ensures efficient inference and can handle sequences of arbitrary length with reduced inference cost. Its performance across a wide range of benchmarks makes it a promising solution for our sub-tasks, given its extensive task capabilities and superior performance in baseline benchmarks compared to similar or larger language models. While we considered experimenting with its larger variant, Mixtral-8x7b (Jiang et al., 2024a), limitations on available resources forced us to deal in depth only with the small variant, Mistral-7b.

Llama 2 (Touvron et al., 2023b): A language model that represents a significant advancement in natural language processing. It is a collection of pre-trained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters. With its large parameter count and advanced architecture, Llama 2 is designed to tackle complex language understanding and generation tasks effectively. It outperforms many other mod-

els, including its predecessor, Llama 1, in various benchmarks, demonstrating superior capabilities in reasoning, mathematics, and code generation. Leveraging its extensive parameterization and innovative techniques, Llama 2 offers state-of-the-art performance across a wide range of NLP applications, making it a notable contender in the field. For our experiments we were able to experiment with various configurations wit the 7 billion and the 13 billion models. Our involvement with the 70 billion parameter model has been restricted due to limitations associated with the extensive parameter count, particularly during the fine-tuning process.

Phi-2 (Gunasekar et al., 2023): An advanced language model designed to address complex natural language processing tasks efficiently. It is part of the small language models (SLMs) released by Microsoft Research team. With its innovative architecture and extensive parameter count, Phi-2 surpasses its predecessor, Phi-1, in various benchmarks, showcasing superior performance in reasoning, comprehension, and text generation. Leveraging cutting-edge techniques and a comprehensive understanding of language patterns, Phi-2 demonstrates remarkable capabilities across a diverse range of NLP applications, solidifying its position as a prominent model in the field. Given its 2.7 billion-parameter architecture, which exhibits exceptional reasoning and language understanding abilities in comparison to various Llama 2 iterations and Mistral-7b, we are confident that this model will deliver noticeable performance for both of our sub-tasks.

B Experimental Setup

In our experiments, we employed the Google Colab platform and Kaggle, leveraging various open-source Python packages such as Transformers, TRL (Transformer Reinforcement Learning) (von Werra et al., 2020), PEFT (Parameter-Efficient Fine-Tuning) (Mangrulkar et al., 2022), BitsAndBytes, Accelerate (Gugger et al., 2022), and Sentence-Transformers.

Encoders BERT-SE[2]: During fine-tuning, a learning rate of $3e^{-5}$ was used, with a batch size of 16 samples processed in each iteration, over the course of 3 epochs. This process aimed to adapt the pre-trained model to better suit our sub-task. Our optimizer was AdamW and our learning scheduler was linear. Same setup was used for the fine-tuning of the BERT encoder.

RoBERTa-WNGRD[3] underwent fine-tuning on the train split of each dataset, utilizing a learning rate of $3e^{-5}$, a batch size of 16, and running for 3 epochs. The opptimizer was also AdamW and the learning scheduler was linear. RoBERTa-large was fine-tuned on the train split of each sub-task's specific dataset using identical configurations.

DeBERTaV3-TS[4], like DeBERTaV3-base, underwent a fine-tuning process similar to the RoBERTa-WNGRD system, differing only in the batch size, which was set to 4.

LLMs Phi-2[5] underwent fine-tuning using the prompt format outlined in Section *Prompting Details*. The fine-tuning process involved setting a learning rate of $2e^{-5}$ and a batch size of 2, with the model trained for 250 steps. We conducted experiments with different configurations of r and lora_alpha, encompassing combinations such as r = 64, 128 and lora_alpha = 64, 128. The dropout rate was consistently set to 0.1 across all experiments. We used an AdamW optimizer and a constant learning scheduler. Despite promising benchmarks accompanying its release, the model's performance during inference on the test split of both sub-tasks' datasets was subpar, scoring lower compared to the encoders mentioned above. This discrepancy raises the possibility, supported by various reports, that the model's training process using methods like quantization and LoRA may not be fully optimized yet, particularly given its recent introduction.

Both variations of Llama 2[6], with 7 billion and 13 billion parameters, underwent the same fine-tuning pipeline described earlier, utilizing the QLoRA technique. The fine-tuning process followed the prompt format outlined in Section D (*Prompting Details*), employing a learning rate of $2e^{-5}$ and a batch size of 1, with each model trained for 250 steps. Despite experimenting with various combinations of values for r and a (32, 64, 128), while the dropout rate was consistently set to 0.1, the results were disappointing. As a text generation model, Llama 2 provided explanations for each multiple-choice prompt. However, even when

[2] https://huggingface.co/JazibEijaz/bert-base-uncased-finetuned-semeval2020-task4b-append-e3-b32-l4e5

[3] https://huggingface.co/DeepPavlov/roberta-large-winogrande

[4] https://huggingface.co/sileod/deberta-v3-large-tasksource-nli

[5] https://huggingface.co/microsoft/phi-2

[6] https://huggingface.co/docs/transformers/en/model_doc/llama2

incorrectly predicting a choice as correct, the generated explanations often lacked logical coherence. Many explanations produced during the inference phase were irrelevant to the context of the brain teaser, indicating a failure to capture the reasoning path of most multiple-choice questions. In summary, both variations of Llama 2, despite their large scale, proved incapable of effectively understanding and reasoning through the multiple-choice questions provided.

The Mistral-7b[7] model outperformed all others significantly. Prior to fine-tuning, we applied the QLoRA technique. Using a learning rate of $2e^{-5}$ and a batch size of 2, each model underwent fine-tuning for 250 steps using the train split of the sub-tasks' dataset. The initial results were promising. During experimentation with the r and a parameters, while maintaining a dropout of 0.1, certain patterns emerged. Specifically, we observed higher quality explanations and scores when using higher rank values, ranging from (16, 32, 64, 128). This outcome was expected, as higher rank values correspond to higher precision weight changes, resulting in superior weight tuning and overall model performance. Interestingly, when the ratio of a/r was low (0.5 - 1), explanations maintained high quality irrespective of predictions, implying a coherent reasoning path even if the predicted choice was incorrect. However, setting the a/r ratio to 2 or 4 potentially enhanced results, signifying a stronger influence from QLoRA layers on the base model. However, this adjustment led to a decline in the quality of explanations. The improvement could be attributed to the model's low intrinsic dimensionality. Despite having many parameters, the effective dimensionality of the model's learned representations is low. Consequently, after conducting several experiments, the best-performing model regarding word puzzles aligns with this concept. After conducting numerous tests, we achieved our best performances with the first model using r=128 and alpha=128, and the third best using r=64 and alpha=32. These models are denoted as Mistral-7b_lora_r_lora_a, representing Mistral-7b_128_128 and Mistral-7b_64_32 configurations, respectively.

Our exploration of Mistral-8x7b[8] was constrained, yet initial results were promising, despite the limited configurations. Further experimentation with various hyperparameter settings may yield improved performance. In our single attempt with this system, we employed a learning rate of $2e^{-5}$ and a batch size of 2, fine-tuning the models for 250 steps using the train split of the sub-task's dataset. Both r and a were set to 128, accompanied by a dropout rate of 0.1. This configuration was selected based on the r and a values of the best-performing model across both sub-tasks, Mistral-7b. Despite its larger scale, Mistral-8x7b achieved the second-best accuracy during inference on the test split regarding the first subtask, trailing behind its smaller variation, Mistral-7b. This model is referenced in the results table of both sub-tasks as Mistral-8x7b_128_128. Further experimentation with various configurations may yield improvements, particularly when leveraging the low intrinsic dimensionality and redundancy inherent in the model.

C QLoRA hyperparameters

Initially, we employed the QLoRA technique (Dettmers et al., 2023) for optimization. The QLoRA technique entails the following steps. First we quantized the models using 4-bit precision to reduce memory usage and computational requirements. The quantization process was facilitated by the BitsAndBytes library. Following quantization, we implemented the LoRA technique (Hu et al., 2021) using the PEFT library. LoRA, applied to the quantized model, resulted in the creation of Quantized LoRA (QLoRA). This pipeline effectively addresses the challenges posed by memory-intensive models on hardware with limited capabilities, ensuring optimized performance and resource utilization. Regarding the hyperparameters of the QLoRA, the rank *(r)* determines the dimensionality of the low-rank approximation used in the adapter layers, while alpha *(a)* is the scaling factor that determines the magnitude of the newly learned weights compared to the original model's weights. The choice of alpha influences how much emphasis is given to the task-specific information compared to the pre-trained knowledge encoded in the original model.

In our experiments, we observed that lower values of r occasionally yielded slightly superior results. This phenomenon can be attributed to the regularization effect introduced by lower-rank approximations. Essentially, lower-rank approximations act as a form of regularization, discouraging

[7] https://huggingface.co/mistralai/Mistral-7B-v0.1

[8] https://huggingface.co/mistralai/Mixtral-8x7B-v0.1

the model from memorizing the training data and instead promoting the learning of more generalizable patterns. This regularization effect becomes particularly significant when dealing with small datasets, as the risk of overfitting is heightened in such scenarios. By limiting the model's capacity through lower-rank approximations, we encourage it to focus on learning essential features and avoid capturing noise or idiosyncrasies present in the training data. Therefore, in our case where the dataset size is small, the regularization provided by lower-rank approximations becomes crucial. It helps prevent overfitting and encourages the model to generalize better to unseen data, ultimately leading to improved performance in certain cases.

Table 7 depicts further analysis of LoRA hyperparameters for Mistral and Mixtral models, which have exhibited the best results among all other models and across the two tasks. Due to computational restrictions, we trained the Mixtral model, which is eight times larger, only for the best performing hyperparameters of Mistral, as a proxy for the performance difference.

D Prompting Details

Here, we provide a comprehensive overview of the prompt utilized consistently throughout the fine-tuning process of the LLMs, which ultimately led to optimal performance across both sub-tasks. Prompt:

```
### Instructions:
Below is an instruction that describes a
multiple choice task. Answer the following
multiple choice question by giving the
most appropriate response. Answer should
be one among options provided after the
question. Select the most suitable answer
while making the necessary assumptions.
Give only answer and a short explanation
of two or three sentences. Nothing else.

### Input:
Question: {question}
1) {a}
2) {b}
3) {c}
4) {d}

### Answer:
The correct answer is: {label}) {answer}
```

In the *Instructions* section, we define the task and provide detailed steps for the system. Results varied depending on the content of the *Instructions* section. It's important to note that our model isn't just tasked with selecting the most appropriate choice from the given options; it's also instructed to generate a brief explanation. This additional step aims to assess the model's ability to identify and comprehend a logical reasoning path that can justify its chosen answers for each multiple-choice problem. Given that the questions are brain teasers that challenge common sense, this approach helps us gauge the model's understanding and reasoning capabilities more effectively. In the *Input* section, we structure the provided dataset into a multiple-choice question format. Each component serves a specific purpose:

Question {question} This section contains the main question extracted from the dataset.

Choices ({a}, {b}, {c}, {d}): These represent the options provided as answers for the question within the dataset.

Correct Answer {label}) {answer} This section indicates the correct label and its corresponding answer from the dataset.

This structured format enables the model to comprehend and process each question along with its associated choices and correct answer during the fine-tuning training process. During the *inference phase*, the same prompt is reproduced, with the sole distinction of a blank space within the Answer section. This deliberate inclusion of a blank space aims to support the model's text generation process. In inference, the model is tasked with generating the correct answer using the information presented in the prompt. This setup enables the model to dynamically generate responses, utilizing its comprehension of the question and the contextual details provided within the prompt.

E Assessment and Insights on Dataset Quality

Upon reviewing our incorrect predictions across both sub-tasks, subsequent to the task organizer releasing the labels for the test split of the datasets, we reached several conclusions. Across all triplets, encompassing original, semantic, and context reconstruction statements, we observe a considerable degree of ambiguity in various patterns. This ambiguity often leads to inconsistent selection of correct answers, even when answered by humans. This underscores the need for clearer formulation of ques-

System	Original	Semantic	Context	Ori. + Sem.	Ori. + Sem. + Con.	Overall
Task A						
Mistral-7b_64_128	**.850**	**.825**	**.775**	**.825**	**.700**	**.817**
Mistral-7b_16_64	.800	.800	.850	.750	.725	.817
Mixtral-8x7b_128_128	.850	.825	.725	.800	.700	.800
Mistral-7b_128_64	.850	.800	.725	.775	.625	.792
Mistral-7b_64_32	.850	.775	.725	.750	.675	.783
Mistral-7b_8_16	.800	.800	.700	.750	.625	.767
Mistral-7b_128_32	.825	.775	.725	.750	.600	.775
Task B						
Mistral-7b_128_128	**.844**	**.844**	**.813**	**.719**	**.625**	**.833**
Mistral-7b_8_16	.781	.938	.781	.719	.562	.833
Mistral-7b_16_16	.812	.812	.875	.688	.625	.833
Mistral-7b_8_8	.875	.812	.812	.750	.688	.833
Mistral-7b_16_32	.875	.812	.781	.750	.594	.823
Mistral-7b_64_32	.844	.875	.719	.750	.562	.812
Mistral-7b_128_64	.844	.812	.781	.688	.531	.812
Mistral-7b_64_64	.719	.812	.625	.625	.406	.719
Mixtral-8x7b_128_128	.625	.719	.625	.531	.375	.656

Table 7: The performance of various LoRA hyperparameters for Mistral and Mixtral in both sub-tasks.

tions and unambiguous expression to enhance the accuracy of model predictions. Another notable pattern we identified pertains to the quality control of semantic reconstruction in certain questions. In these instances, some words were not replaced with accurate synonyms, resulting in a shift in the definition of the brain teaser presented by the question. While this may not inherently be problematic, the dataset's correct answers remained unchanged compared to the original version of the question. This discrepancy suggests that the alteration in question definition went unnoticed by the task organizers, leading to some erroneous predictions by our model, when in reality the correct context of the provided multiple-choice statement was captured by our system. The two observations above highlight the inherent difficulty in generating clear and precise brain teasers, as well as the challenge that models face in understanding them. In the above scenarios, our top-performing model either detects the presence of a contradiction in the questions and opts to select "None of above," as elucidated in its brief and explanatory justification, or it provides an incorrect answer based on the dataset's answer but correctly reflects the problem context, which may have been altered due to inadvertent synonym usage.

DeepPavlov at SemEval-2024 Task 3: Multimodal Large Language Models in Emotion Reasoning

Julia Belikova and **Dmitrii Kosenko**
Moscow Institute of Physics and Technology
belikova.iua@phystech.edu, kosenko.dp@mipt.ru

Abstract

This paper presents the solution of the DeepPavlov team for the Multimodal Sentiment Cause Analysis competition in SemEval-2024 Task 3, Subtask 2 (Wang et al., 2024). In the evaluation leaderboard, our approach ranks 7th with an F1-score of 0.2132. Large Language Models (LLMs) are transformative in their ability to comprehend and generate human-like text. With recent advancements, Multimodal Large Language Models (MLLMs) have expanded LLM capabilities, integrating different modalities such as audio, vision, and language. Our work delves into the state-of-the-art MLLM Video-LLaMA, its associated modalities, and its application to the emotion reasoning downstream task, Multimodal Emotion Cause Analysis in Conversations (MECAC). We investigate the model's performance in several modes: zero-shot, few-shot, individual embeddings, and fine-tuned, providing insights into their limits and potential enhancements for emotion understanding.

1 Introduction

In the dynamic domain of artificial intelligence, the emergence of MLLMs has gained significant interest due to integrating input from different modalities, such as audio, vision and language, opens up in-depth perceptual and interpretive capabilities in dialogues instead of chat-based dialogue systems (Konovalov et al., 2016).

These models exhibit impressive potential in resolving a lot of challenges and have been deployed across various sectors, including banking support systems, social services, and as adjuncts in psychological assistance. In these applications, the deciphering of user intent and emotions is crucial for generating pertinent responses. Consequently, one of the most important domains within MLLM research is emotion reasoning.

This paper explores a specific facet of emotion reasoning: Multimodal Emotion Cause Analysis in Conversations (MECAC) (Wang et al., 2024). This task involves emotion recognition and matching emotional states with their causes in the context of a conversation, leveraging inputs from different modalities such as text, audio, video and more.

Despite the remarkable advancements in MLLMs, their capabilities and limitations in the high-potential area of emotion reasoning remain active topics for research. One of the seminal contributions to the investigation of MLLMs capabilities within the area of emotional reasoning is delineated in (Lian et al., 2023). The authors introduce a novel task, Explainable Multimodal Emotion Reasoning (EMER), and proceed to evaluate the efficacy of modern multimodal models in addressing EMER. Their research focuses on the integration and interpretability of emotional cues across diverse modalities, thereby advancing the understanding of emotion reasoning.

To address the described issue, we propose to continue research of the MLLMs capabilities in emotion reasoning by evaluating one of the most promising models, Video-LLaMA (Zhang et al., 2023), also explored in Lian et al. (2023), for MECAC on the Emotion-Cause-in-Friends dataset.

The work evaluates the performance of the model in three modes:

1. *Zero-shot and Few-shot modes*. These modes are utilized to evaluate the model's initial capabilities in emotion reasoning.

2. *Individual Embeddings mode*. In this mode, embeddings from individual modalities are employed alongside trained basic heads to address MECAC.

3. *Fine-tuned mode*. This mode is used to evaluate specialized emotion reasoning capabilities.

Experimental results demonstrate the considerable potential of MLLMs in the domain of emotion reasoning.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1747–1757

2 Related Work

2.1 Multimodal Large Language Models

The ascent of Large Language Models such as LLaMA2 (Touvron et al., 2023), Qwen (Bai et al., 2023), Mistral (Jiang et al., 2023) has marked a significant milestone in the field of artificial intelligence. These models have demonstrated exceptional capabilities in language reasoning and decision-making, closely mirroring human-level performance.

The integration of adapters to align pre-trained encoders from different modalities with textual LLMs has given rise to a new class of MLLMs such as: Flamingo (Alayrac et al., 2022), BLIP-2 (Li et al., 2023a), MiniGPT-4 (Zhu et al., 2023), mPLUG-Owl (Ye et al., 2023), VideoChat (Li et al., 2024a), InstructBLIP (Dai et al., 2023), Video-ChatGPT (Maaz et al., 2023), Video-LLaVA (Lin et al., 2023), VideoChat2 (Li et al., 2024b), Video-LLaMA (Zhang et al., 2023) (Table 1).

Model	Modality
Flamingo	I, V, T
BLIP-2	I, T
MiniGPT-4	I, T
mPLUG-Owl	I, V, T
InstructBLIP	I, T
Video-ChatGPT	I, V, T
VideoChat2	I, V, T
Video-LLaMA	I, V, A, T
Video-LLaVA	I, V, T

Table 1: Multimodal Large Language Models. T, A, I, and V stand for text, audio, image and silent video, respectively

These models have gained impressive results in well-known general domains (Wu et al., 2023): temporal perception and reasoning, casual inference, and spatial perception and analysis.

2.2 Emotion Reasoning in Conversation

Our work delves into MECAC, a derivation of ECPE (Xia and Ding, 2019), the downstream task of emotion reasoning. Given a conversation sequence consisting of N utterances, $U = \{U_1, U_2, \ldots, U_N\}$, where each utterance U_i is accompanied by a corresponding speaker identity, textual content, and an associated audio-visual clip.

The task is to output a set of emotion-cause pairs $E = \{(e_i, c_i)\}_{i=1}^{M}$, where each pair contains:

- e_i: an emotion utterance U_j that expresses an emotion.
- c_i: a cause utterance U_k that is identified as the cause of the emotion expressed in U_j.

Additionally, each emotion utterance e_i is tagged with an emotion category EC from a predefined set of emotion categories $EC = \{EC_1, EC_2, \ldots, EC_K\}$.

The exploration of emotion reasoning within the context of conversations has traditionally been addressed using various classical approaches. Recurrence-based or graph-based methods have been particularly popular due to their ability to capture sequential and relational data effectively. Notable methods in this domain include: MC-ECPE-2steps (Wang et al., 2023), which focuses on two-step recurrence-based emotion-cause pair extraction; Joint-GCN (Li et al., 2023b), which leverages recurrent and graph convolutional networks for joint emotion-cause detection; ECQED (Zheng et al., 2023), which extends the emotion-cause pair extraction to a quadruple extraction task and structural and semantic heterogeneous graph for conversation representation; CORECT (Nguyen et al., 2023), which enhances conversation understanding through relational temporal graph neural networks; and COGMEN (Joshi et al., 2022), which utilizes contextualized graph neural networks for multimodal emotion recognition.

In this paper, we investigate the capabilities of MLLMs for solving MECAC by evaluating the state-of-the-art model, Video-LLaMA. It is pertinent to acknowledge the application of MLLMs in a variety of sentiment and emotion recognition (Aslam et al., 2023), in the domain of EMER. Previous works indicate that MLLMs demonstrate notable efficacy in these complex tasks, which highlights their potential for advancing the frontier of emotion reasoning research.

2.3 Video-LLaMA architecture

Next, we summarize the key points of Video-LLaMA's architecture.

Video-LLaMA is a multimodal framework designed to extend the capabilities of frozen LLMs by enabling them to process and respond to audio-visual content.

Visual Encoder. The visual encoding component employs a pre-trained image encoder to compute representations from individual frames of a video. It introduces a frame embedding layer to

provide temporal information and incorporates a video Q-former to generate visual query tokens that encapsulate the temporal dynamics of visual scenes. A linear layer is introduced to transform video embedding vectors into query vectors that are compatible with the embedding space of LLMs.

Audio Encoder. For audio processing, Video-LLaMA leverages ImageBind (Girdhar et al., 2023). It also uses a similar architecture to the visual encoder to obtain audio embeddings for the LLM module.

Cross-Modal Training. The training process involves multi-branch, cross-modal pre-training to achieve both vision-language and audio-language alignment. The vision-language pre-training includes a video-clips-to-text generation task and static image-caption learning. The audio-language pre-training leverages the audio encoder and vision-text data to align with the LLM's embedding space.

Standard Inference. During inference, Video-LLaMA is capable of zero-shot video and audio understanding. It processes video frames and audio signals, converts them into query representations that are concatenated with textual input embeddings of LLMs, and generates responses grounded in the video's visual and auditory content.

3 Methods

3.1 Zero-shot and Few-shot

Today's LLMs are developed using extensive datasets and are further fine-tuned to comprehend and follow instructions, granting them the capacity to perform certain tasks in a zero-shot fashion (Tirskikh and Konovalov, 2023). Investigating how these capabilities are exhibited in multimodal models represents an active area of research.

To evaluate the capabilities of Video-LLaMA in the zero-shot emotion reasoning subtasks, we use structured templates such as the one detailed in Appendix A, Listing 4

While LLMs demonstrate remarkable zero-shot capabilities, they still fall short on more complex tasks within the zero-shot setting. Consequently, it is essential to evaluate their few-shot capabilities as well. To achieve this, we employ prompt templates, such as the one described in Appendix A, Listing 5.

3.2 Individual Embeddings

In this work, we also investigate the capabilities of embeddings obtained from the output of Video-LLaMA. Specifically, we extract multimodal embeddings corresponding to each fragment of the conversation. These embeddings are derived from the last semantic token of the last hidden state during the generation of responses to prompts formatted as shown in Listing 1.

```
# First option
<Item Value> Describe the behavior of
the speaker in this <Item Name> in one
word:
# Second option
<Item Value> Describe what is happening
in this <Item Name> in one word:
# Third option
<Item Value> Describe the emotional
state of the speaker in this <Item Name>
in one word:
```

Listing 1: Prompt templates for multimodal embeddings generation

To utilize these embeddings for our task, we integrate classical heads based on Multi-Layer Perceptron (MLP), Bidirectional Long Short-Term Memory (BiLSTM), and Self-Attention mechanisms. In the context of MLP, we adopt a straightforward approach for multi-class classification of emotions and binary classification of causes. Importantly, for the binary classification of causes, we consider all possible pairs of utterances. The probability that a pair belongs to a specific class is computed based on the output from the linear layer, which receives a concatenated representation of the utterance pairs as its input.

The BiLSTM head is implemented similarly to the MLP. For the self-attention mechanism, we employ multiple layers of a classical architecture. For both approaches — the MLP and BiLSTM-based heads — we utilize Cross-Entropy as the loss function.

It is also worth noting that there is a class imbalance in the case of binary causes classification. According to the authors of the dataset about 55.73% of the utterances are annotated with one of the six basic emotions, and 91.34% of the emotions are annotated with the corresponding causes in the ECF dataset. As a result, the matrices of some conversations divided into utterances become quite sparse. To mitigate the impact of imbalance, we propose several balancing methods: simple weighting of the loss function and adaptive weighting. In the first case, a constant scaling factor is chosen to increase the influence of the minority class, while in the

second case, balancing is done for each individual batch based on the current class frequency.

3.3 Fine-tuning

The fine-tuning stage employs Low-Rank Adaptation (LoRA) (Hu et al., 2021) to modify the pre-trained parameters of the LLaMA module within Video-LLaMA, while the visual and audio encoders remain unchanged. We design prompts for fine-tuning, outlined in Listings 2 and 3, that closely align with the format proposed in (Lei et al., 2023).

```
You are expert of multimodal emotion
classification and emotion cause
recognition.

The following is a conversation that
involves several speakers.

Here is a conversation that is described
in several fragments and includes
subtitles, video, and audio:

Utterance_1
<Speaker Name>: <Speaker Text>
Video: <Video>
Audio: <Audio>
...

Select the emotion label of each
utterance from <neutral, surprise,
fear, sadness, joy, anger, disgust>
and predict the ids of utterances that
caused this emotion.
```

Listing 2: Instruction format for the fine-tuning stage

```
Utterance_1
Emotion: <Emotion>
Causes: 1
...
```

Listing 3: Response format for the fine-tuning stage

4 Experiments

For the experiments described below, we use the Emotion-Cause-in-Friends (ECF) dataset (the official train part), which is divided into train, validation, and test sets in accordance with the proportion 8:1:1. We used train part due to test split is not officially available for extensive experiments.

4.1 Zero-shot and Few-shot

In the zero-shot experiments, the model exhibits a loss of ability to follow general instructions and ceases responding to the guidelines provided, instead demonstrating a tendency for a detailed description of the events observed in the video. Examples of this behavior are visible in the experimental data presented in Appendix B. In few-shot experiments with Video-LLaMA, we observe the same pattern.

4.2 Individual Embeddings

Metrics. In evaluating the model's performance on the emotion classification subtask, we utilize two principal metrics: the macro F1-score, which provides a balanced measure of precision and recall across all classes, and Accuracy, reflecting the overall proportion of correctly identified instances. For the causal classification subtask, we similarly measure performance using the binary F1-score, which is tailored to binary classification problems, alongside Accuracy to determine the proportion of true results in the dataset.

Training configuration. Each training session is run in 50 epochs. For emotion classification, 32 utterances are used as one batch. For cause classification, one batch describe one conversation and an accumulation of 6 batches for gradient optimization is used.

To address the challenges presented by MECAC, our approach encompassed two distinct training schemas: joint and separate training for the dual classification objectives, namely emotions and causes. The joint training final loss function was composed as a linear combination of the individual losses from both classification heads as in MTL systems (Karpov and Konovalov, 2023).

Initial observations from the joint training indicated that the combination of loss functions from the emotion and cause components was instrumental in enhancing the model's generalization capabilities. However, this joint strategy appeared to reach a plateau, failing to deliver the maximum attainable performance in the later stages of training.

Further experimentation yielded additional insights, particularly in the domain of model convergence. For the emotion classification task, the MLP head emerged as the superior architecture, leading to the most optimal model convergence. Conversely, the BiLSTM head demonstrated a marked advantage in the cause classification domain.

Also, as mentioned above for the training of cause classification it is suggested to perform balancing of the loss function. In practice, the assumption to mitigate class imbalance has proven to be highly significant. According to the experimental results, the best convergence was provided by the use of a constant weight coefficient, notably a value of 3, to give greater emphasis to predictions of the minor classes within the loss function.

Prompt optimization. We evaluate three distinct prompt configurations to derive embeddings for each modality under consideration. The experimental results reinforce the notion of textual content as a leading modality, with the third prompt configuration demonstrating particular efficacy. Accordingly, the tables in Appendix C present the training results as evaluated on the test subset, including all combinations of the prompts applied to the audio and video modalities.

Modality impact. To evaluate the contribution of each modality to the overall effectiveness of the classification tasks, we conducted several experiments. The validation results are depicted in Figure 1, and the test results confirm the observed trend. The text modality emerges as the most influential, exerting the greatest effect on the model's predictive accuracy. In a secondary position, the audio modality is found to have a considerable impact, albeit less than that of text. The video modality, while still contributing to the overall model performance, is observed to have the least influence among the three.

The leading role of textual modality can also be substantiated by the information provided by the ECF authors, who state that approximately 8% of the emotion causes in the dataset are the events mainly reflected in the acoustic or visual modalities. It's also important to note the least valuable modality in these experiments: the visual modality. We suppose that this is justified by the lack of confidence of the models in visual feature space, which, most likely, can be eliminated by fine-tuning of the visual encoding branch.

4.3 Fine-tuning

In the fine-tuning phase of our experiments, we employ LoRA technique to fine-tune the parameters of the language model component, specifically, the 4-bit quantized Llama-2 7b model, it's selected due to resource constraints. We configure LoRA with an alpha value of 16 and a low-rank factor of 8, while a dropout rate of 0.1 is utilized to prevent overfitting. Our method focuses on selectively adapting only the self-attention projection modules within the transformer architecture. This refines the model's focus on salient features for the tasks at hand without necessitating a comprehensive retraining of the entire network. Training batches are set to a size of 4, and the fine-tuning process is conducted over a single epoch, covering the full training dataset.

The fine-tuning strategy yields notable improvements in the model's performance across two distinct classification tasks. For the emotion classification task, the model achieves a macro F1-score of 0.6500 and an Accuracy of 0.7412. This represents a significant enhancement in the model's ability to discern and categorize emotional content within the input data accurately. In the causal classification task, the model demonstrates a binary F1-score of 0.3824 and an Accuracy of 0.9220. While the binary F1-score appears modest, the high Accuracy underscores the model's effectiveness in identifying causal relationships within the tested dataset.

5 Conclusion

In this paper, we conduct an analysis of the MLLM Video-LLaMA with an emphasis on its emotion understanding capabilities in MECAC. Our experiments show that multimodal models, in their current iteration, exhibit limitations in deciphering emotional states under zero-shot and few-shot modes.

To enhance the capabilities of such models in emotion understanding, our findings indicate that task-specific dataset fine-tuning is an essential step. Despite the challenges observed, the raw embeddings generated by the Video-LLaMA model show promising potential as a foundation for improving emotion recognition performance.

The implications of this research highlight the necessity for continued development and refinement of multimodal learning frameworks. Future work may concentrate on expanding the diversity of the datasets used for fine-tuning to include a broader spectrum of emotional expressions and cultural contexts. This could mitigate existing biases and enhance the model's generalizability across various demographics and scenarios. Moreover, incorporating advanced techniques such as transfer learning and domain adaptation could further enhance the model's proficiency in interpreting nuanced emotional states (Chizhikova et al., 2023).

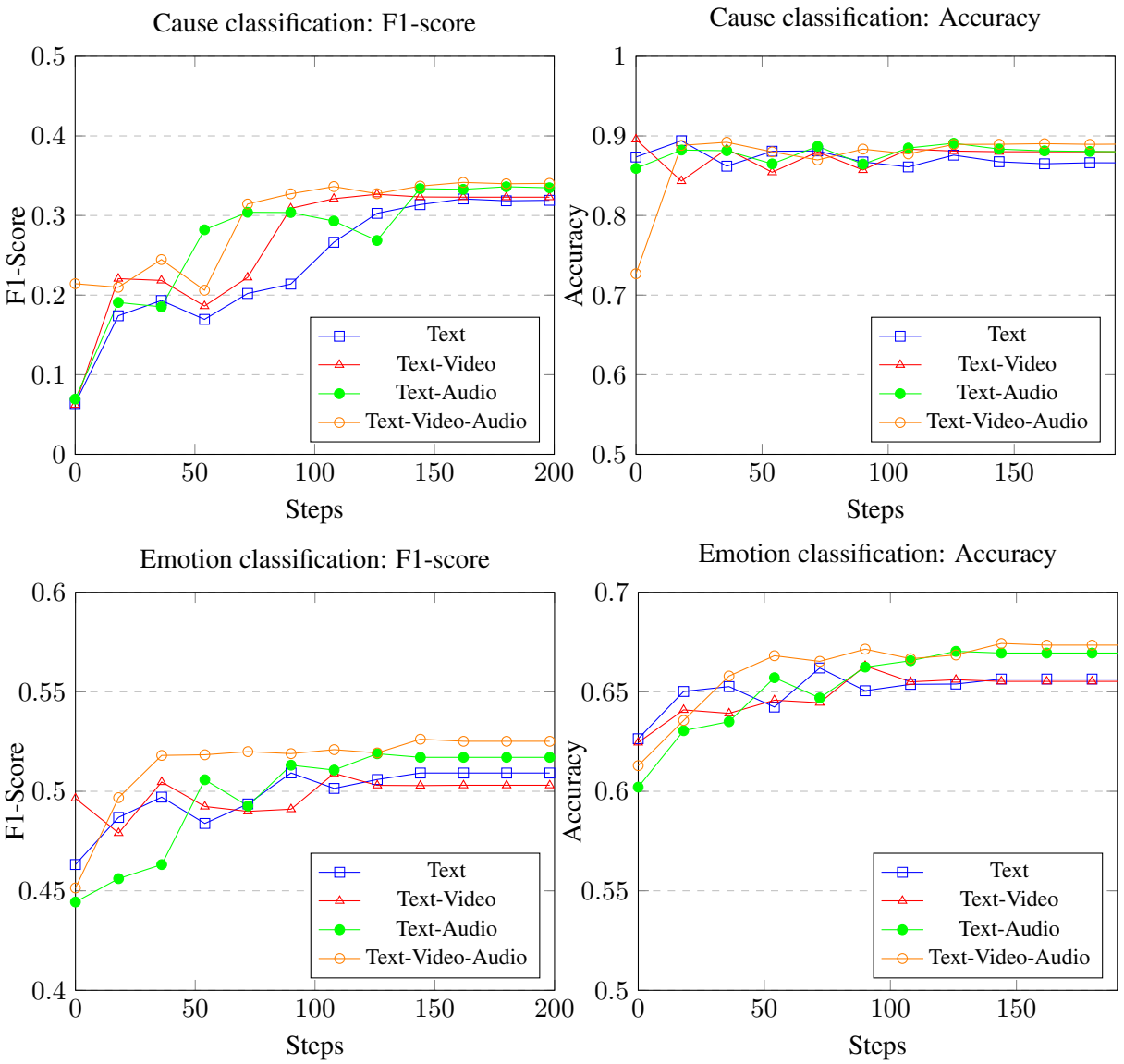


Figure 1: Impact of different modalities on the classification tasks

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning.

Ajwa Aslam, Allah Bux Sargano, and Zulfiqar Habib. 2023. Attention-based multimodal sentiment analysis and emotion recognition using deep neural networks. *Applied Soft Computing*, 144:110494.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Anastasia Chizhikova, Vasily Konovalov, and Mikhail Burtsev. 2023. Multilingual case-insensitive named entity recognition. In *Advances in Neural Computation, Machine Learning, and Cognitive Research VI*, pages 448–454, Cham. Springer International Publishing.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang,

Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Abhinav Joshi, Ashwani Bhat, Ayush Jain, Atin Vikram Singh, and Ashutosh Modi. 2022. Cogmen: Contextualized gnn based multimodal emotion recognition.

Dmitry Karpov and Vasily Konovalov. 2023. Knowledge transfer between tasks and languages in the multi-task encoder-agnostic transformer-based models. In *Computational Linguistics and Intellectual Technologies*, volume 2023.

Vasily Konovalov, Oren Melamud, Ron Artstein, and Ido Dagan. 2016. Collecting Better Training Data using Biased Agent Policies in Negotiation Dialogues. In *Proceedings of WOCHAT, the Second Workshop on Chatbots and Conversational Agent Technologies*, Los Angeles. Zerotype.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2024a. Videochat: Chat-centric video understanding.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark.

Wei Li, Yang Li, Vlad Pandelea, Mengshi Ge, Luyao Zhu, and Erik Cambria. 2023b. Ecpec: Emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1754–1765.

Zheng Lian, Licai Sun, Mingyu Xu, Haiyang Sun, Ke Xu, Zhuofan Wen, Shun Chen, Bin Liu, and Jianhua Tao. 2023. Explainable multimodal emotion reasoning.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models.

Cam Van Thi Nguyen, Tuan Mai, Son The, Dang Kieu, and Duc-Trong Le. 2023. Conversation understanding using relational temporal graph neural networks with auxiliary cross-modality interaction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Danil Tirskikh and Vasily Konovalov. 2023. Zero-shot ner via extractive question answering. In *Advances in Neural Computation, Machine Learning, and Cognitive Research VII*, pages 22–31, Cham. Springer Nature Switzerland.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Heqing Ma, Rui Xia, Jianfei Yu, and Erik Cambria. 2024. Semeval-2024 task 3: Multimodal emotion cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2022–2033, Mexico City, Mexico. Association for Computational Linguistics.

J. Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu. 2023. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256, Los Alamitos, CA, USA. IEEE Computer Society.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-owl: Modularization empowers large language models with multimodality.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding.

Li Zheng, Donghong Ji, Fei Li, Hao Fei, Shengqiong Wu, Jingye Li, Bobo Li, and Chong Teng. 2023. Ecqed: Emotion-cause quadruple extraction in dialogs.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models.

A Appendix

```
Video: <Conversation Video>
Speaker: <Speaker-1 Name>
Text: <Speaker-1 Text>
...
Speaker: <Speaker-i Name>
Text: <Speaker-i Text>

Predict last emotion of <Speaker Name-i>.
Choose emotion from this list: surprise, fear, sadness, neutral, joy, anger,
disgust.
```

Listing 4: Zero-shot prompt template

```
Video: <Conversation Video>
Speaker: <Speaker-1 Name>
Text: <Speaker-1 Text>
Emotion: <Speaker-1 Emotion>
...
Speaker: <Speaker-i Name>
Text: <Speaker-i Text>
Emotion: <Speaker-i Emotion>

Predict last emotion of <Speaker-i Name>.
Choose emotion from this list: surprise, fear, sadness, neutral, joy, anger,
disgust.
```

Listing 5: Few-shot prompt template

B Appendix

```
Speaker: Joey
Text: Let it go, Ross.
Speaker: Ross
Text: Yeah, well, you did not know Chi Chi.
Speaker: Monica
Text: Do you all promise?
Speaker: All
Text: Yeah! We promise! We will be good!
Emotion: _____

Describe the last emotion of all. Choose from the list: surprise, fear, sadness,
neutral, joy, anger, disgust.
```

Listing 6: Prompt for checking zero-shot capabilities

```
First, the video features a man, woman, and child sitting in a living room with a
green glass table, couch, and a flower vase. The mother is holding a blue blanket
while the man is eating a bowl of cereal. Next, the video switches to a shot of the
 mother holding a yellow flower while the daughter is holding a yellow tube and
talking on her phone. The mother is also seen holding a yellow flower in a
different shot.

The video then shows a couple holding a cake in a kitchen, with a wooden table and
chairs. The couple is standing in front of a counter with a white bowl and a white
ceramic plate. Then, the video shows a man, woman, and child sitting in a living
room with a flower vase on a green glass table, a couch, and a chair. The man and
woman are eating cereal while the child is holding a yellow tube and talking on the
 phone.

The video then shows a shot of the mother, daughter, and son sitting in a living
room with a green glass table, couch, and a chair. The mother is holding a blue
blanket while the son is eating cereal. Finally, the video features the mother
holding a blue blanket while the daughter is holding a yellow tube and talking on
her phone.
```

Listing 7: Model response for zero-shot prompt

C Appendix

Prompt format	macro F1-score	Accuracy
Audio-1 Video-1	0.4983	0.6554
Audio-1 Video-2	0.4992	0.6560
Audio-1 Video-3	0.5057	0.6568
Audio-2 Video-1	0.5102	0.6703
Audio-2 Video-2	**0.5251**	**0.6734**
Audio-2 Video-3	0.5131	0.6720
Audio-3 Video-1	0.5010	0.6566
Audio-3 Video-2	0.5105	0.6541
Audio-3 Video-3	0.5078	0.6575

Table 2: Prompt optimization results for emotion classification, where Modality-i is an i-th prompt option

Prompt format	F1-score	Accuracy
Audio-1 Video-1	**0.3505**	**0.8898**
Audio-1 Video-2	0.3496	0.8872
Audio-1 Video-3	0.3494	0.8850
Audio-2 Video-1	0.3480	0.8743
Audio-2 Video-2	0.3327	0.8735
Audio-2 Video-3	0.3194	0.8755
Audio-3 Video-1	0.3360	0.8806
Audio-3 Video-2	0.3325	0.8739
Audio-3 Video-3	0.3184	0.8799

Table 3: Prompt optimization results for cause classification, where Modality-i is an i-th prompt option

iREL at SemEval-2024 Task 9: Improving Conventional Prompting Methods for Brain Teasers

Harshit Gupta, Manav Chaudhary, Tathagata Raha,
Shivansh Subramanian and Vasudeva Varma
International Institute of Information Technology, Hyderabad (IIIT-H)
{harshit.g, manav.chaudhary, tathagata.raha, shivansh.s}@research.iiit.ac.in
vv@iiit.ac.in

Abstract

This paper describes our approach for SemEval-2024 Task 9: BRAINTEASER: A Novel Task Defying Common Sense. The BRAINTEASER task comprises multiple-choice Question Answering designed to evaluate the models' lateral thinking capabilities. It consists of Sentence Puzzle and Word Puzzle subtasks that require models to defy default commonsense associations and exhibit unconventional thinking. We propose a unique strategy to improve the performance of pre-trained language models, notably the Gemini 1.0 Pro Model, in both subtasks. We employ static and dynamic few-shot prompting techniques and introduce a model-generated reasoning strategy that utilizes the LLM's reasoning capabilities to improve performance. Our approach demonstrated significant improvements, showing that it performed better than the baseline models by a considerable margin but fell short of performing as well as the human annotators, thus highlighting the efficacy of the proposed strategies. We have made our code open-sourced for the replicability of our methods. [1]

1 Introduction

Human cognition is characterized by two distinct modes of thinking: vertical and lateral (Waks, 1997). Vertical thinking, often called logical or convergent thinking, follows a structured analytical process based on reasoning and established rules. By contrast, lateral thinking, or "thinking outside the box," is a creative and divergent process that challenges conventional assumptions and explores novel perspectives.

Vertical thinking and lateral thinking are both complementary (Dingli, 2008). Vertical thinking is known for its selectivity and focus, while lateral thinking is known for its creativity and ability to generate alternative approaches and perspectives. It is crucial to recognize the value of both types of thinking and utilize them in a balanced manner to achieve optimal results.

We work on two subtasks (Sentence Puzzle and Word Puzzle) introduced as part of the SemEval-2023 Task 9: BRAINTEASER: A Novel Task Defying Common Sense (Jiang et al., 2024). The Sentence-type brain teasers contain puzzle-defying common sense teasers centred around sentence snippets. For instance, the question "What has a bed but no head, a mouth but no teeth?" challenges the default association of beds with people and forces the solver to consider other possibilities, such as a river (which has a riverbed but no heads or teeth). In Word-type brain teasers, the answer violates the default meaning of the word and focuses on the letter composition of the target question. For example, the question "What word becomes shorter when you add two letters to it?" challenges the assumption that adding letters to a word would make it longer and requires the solver to recognize that the word "short" becomes "shorter" when "er" is added to it.

We used the Gemini 1.0 Pro Model (Team et al., 2023) to evaluate the model's performance in zero-shot and few-shot settings. We also make notable enhancements in the few-shot setting. Firstly, by employing contextualized question selection, we ensure that the model is exposed to more relevant examples by identifying questions from the training set that closely resemble those in the test set. Secondly, we enable the model to generate explanations for correct answer choices during training through reason generation, thereby deepening its comprehension of the examples. These approaches have demonstrated improvements in the evaluation scores.

2 Related Work

Martinez-Gil et al. (2019), Hendrycks et al. (2021),

[1] https://github.com/TheAthleticCoder/iREL-at-SemEval-2024-Task-9.git

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1758–1766

and Singhal et al. (2023) demonstrate recent advancements in multiple-choice question answering. They achieve this by developing new datasets and evaluating large language models (LLMs) on them, thus contributing significantly to the field's progress.

Xie et al. (2023) presents OlaGPT, an innovative framework designed to enhance the reasoning capabilities of large language models (LLMs) by drawing inspiration from human cognitive architecture. OlaGPT integrates cognitive modules such as attention, memory, reasoning, and learning, emphasizing a reasoning module that simulates human-like thought processes. The module then enables OlaGPT to create multiple agents and utilize various thinking templates, including lateral and integrative thinking, to solve reasoning problems effectively.

Huang et al. (2023) introduces a novel evaluation benchmark to assess a model's lateral thinking abilities in an interactive framework, utilizing Lateral Thinking Puzzles as the context.

Meng et al. (2024) proposes a divide-and-conquer approach to LLM reasoning. It involves categorizing questions into subsets based on statistical confidence scores (CS), followed by targeted interventions such as Prior Knowledge-based Reasoning (PKR) and Filter Choices-based Reasoning (FCR) to address nuanced and demanding tasks.

3 Data

The primary dataset (Jiang et al., 2023a) used in this study encompasses data pertinent to two subtasks: sentence puzzles and word puzzles. The puzzle is presented in a single correct MCQ format, where each puzzle consists of a question and several options. Among these options, only one is the correct answer to the puzzle. Creating multiple-choice questions challenges balancing fairness and intellectual engagement (Ma et al., 2021). This necessitates carefully curating distractors that are not only incorrect but also sufficiently challenging. It is worth noting that within the training data for both puzzles, every brainteaser was accompanied by two distractor options alongside the correct option. Please refer to Table 1 for specific sample numbers.

Table 1: Dataset Details

Type of Puzzle	Train Samples	Test Samples
Sentence Puzzle	507	120
Word Puzzle	396	96

4 Methodology

4.1 Zero-Shot Prompting

Initially, we conducted experiments using a zero-shot approach(Brown et al., 2020). In this method, we presented the model with questions and their multiple-choice options and asked it to identify the correct option. Subsequently, we improved the zero-shot approach by introducing the model to a sentence or word puzzle concept depending on the specific subtask under evaluation. We provided the model with the puzzle definition and then asked it to select the correct option from the choices. This modified zero-shot prompt template can be found in App.A.

4.2 Few-Shot Prompting

To enhance the model's performance, we implemented a few-shot prompting technique (Brown et al., 2020). This methodology involved presenting a variable number of examples to the model to facilitate in-context learning for the lateral thinking task. Subsequently, the model was prompted to identify the correct option among the provided choices. App.B shows examples of few-shot prompt templates.

4.3 Contextualized Example Selection

The method outlined above relies on a fixed set of examples for model prompting. We modify this by applying a Dynamic Few-Shot prompting approach inspired by Nori et al. (2023), which enables in-context learning. This method involves selecting samples from the train data that closely match the semantic content of the samples posed in the test data. Initially, all questions from both the training and testing datasets undergo encoding using BERT-Large (Devlin et al., 2019), a pre-trained transformer-based language model. Subsequently, Cosine Similarity is utilized to calculate similarity scores between each question in the testing dataset and all questions in the training dataset.

Based on these similarity scores, we select the $top - n$ most similar examples from the training dataset for each question in the testing dataset, where n varies based on the desired number of

examples to be used in the prompt. This dynamic selection process aims to leverage more relevant examples from the training data to allow for better in-context learning while evaluating each test sample.

4.4 Self-Generated Reasoning

In Section 3 of this paper, we discuss how our training data contains the correct option and two distractor options for each question. We utilize this information to prompt the Gemini Model(Team et al., 2023) and GPT-4 Model (Achiam et al., 2023) to produce detailed reasoning about why the correct option is correct and why the distractor options are incorrect, highlighting potential confusion for test-takers. We include the models' reasoning and examples from the training data during inference on the testing data. This approach aims to improve the quality and precision of the models by providing detailed insights into the reasoning behind the options making up the example. The prompt template used to generate the reasons and some examples of model-generated reasons for the samples from the training data are provided in App.C,D.

4.5 Model and Hyperparameters

The Gemini Pro 1 Model (Team et al., 2023) was used as the primary model in this study. The temperature parameter(Brown et al., 2020) was set to 0.1, to guide the model to indicate its belief regarding the correct option based solely on its pre-existing knowledge base. This low-temperature setting was chosen to minimize the generation of creative or unexpected outputs. Additionally, we set both the top_p and top_k parameters (Brown et al., 2020) to 1. By restricting the model in this manner, we aimed to maintain the relevance and coherence of the responses within the context of our research tasks.

5 Results

The results of our experiments and ablation studies conducted for the Sentence Puzzle and Word Puzzle subtasks are presented in Tables 2 and 3, respectively. The baseline scores from the best-performing model in the Instruction, Commonsense, and Human categories, as provided by Jiang et al. (2023b), serve as benchmarks for comparison. Notably, across all six columns of scores, the Zero-Shot approaches and Few-Shot approaches outperform the baseline Chat-GPT 0 Shot and Roberta-L models by a significant margin.

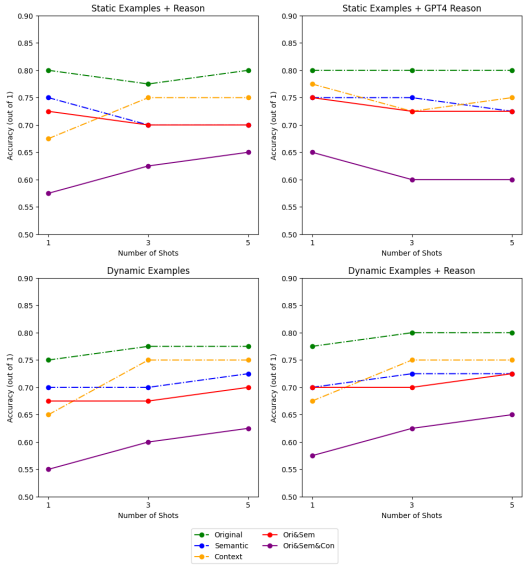


Figure 1: Few-shot prompting performance on the Sentence Puzzle subtask

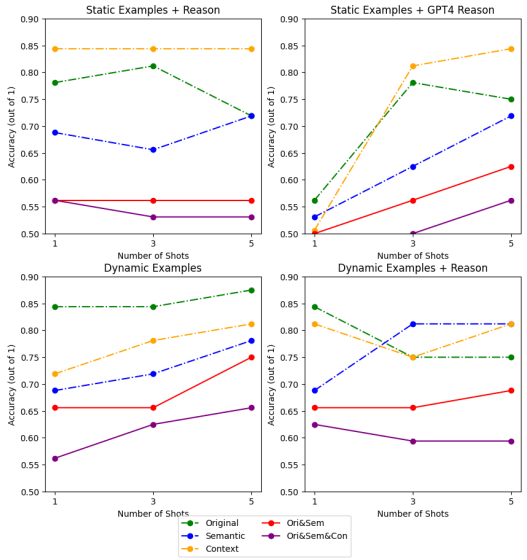


Figure 2: Few-shot prompting performance on the Word Puzzle subtask

Table 2: Results of the Sentence Puzzle subtask: Ori = Original, Sem = Semantic, Con = Context, SE = Static Examples, DE = Dynamic Examples, GPTR = GPT4 Reasoning

Strategy	Ori	Sem	Con	Ori & Sem	Ori & Sem & Con	Overall
Baseline						
Chat-GPT 0 shot	0.6077	0.5933	0.6794	0.5072	0.3971	0.6268
Roberta-L	0.4354	0.4019	0.4641	0.3301	0.2010	0.4338
Human	0.9074	0.9074	0.9444	0.9074	0.8889	0.9198
Zero-Shot						
Direct Prompt	0.775	0.725	0.575	0.700	0.525	0.692
Definition Prompt	0.775	0.725	0.700	0.700	0.575	0.733
Few-Shot						
1 Shot + SE + Reason	**0.800**	**0.750**	0.675	0.725	0.575	0.742
3 Shot + SE + Reason	0.775	0.700	0.750	0.700	0.625	0.742
5 Shot + SE + Reason	**0.800**	0.700	0.750	0.700	**0.650**	0.750
1 Shot + SE + GPTR	**0.800**	**0.750**	**0.775**	**0.750**	**0.650**	**0.775**
3 Shot + SE + GPTR	**0.800**	**0.750**	0.725	0.725	0.600	0.758
5 Shot + SE + GPTR	**0.800**	0.725	0.750	0.725	0.600	0.758
1 Shot + DE	0.750	0.700	0.650	0.675	0.550	0.700
3 Shot + DE	0.775	0.700	0.750	0.675	0.600	0.742
5 Shot + DE	0.775	0.725	0.750	0.700	0.625	0.750
1 Shot + DE + Reason	0.775	0.700	0.675	0.700	0.575	0.717
3 Shot + DE + Reason	**0.800**	0.725	0.750	0.700	0.625	0.758
5 Shot + DE + Reason	**0.800**	0.725	0.750	0.725	**0.650**	0.758

Table 3: Results of the Word Puzzle subtask: Ori = Original, Sem = Semantic, Con = Context, SE = Static Examples, DE = Dynamic Examples, GPTR = GPT4 Reasoning

Strategy	Ori	Sem	Con	Ori & Sem	Ori & Sem & Con	Overall
Baseline						
Chat-GPT 0 shot	0.5610	0.5244	0.5183	0.4390	0.2927	0.5346
Roberta-L	0.1951	0.1951	0.2317	0.1463	0.061	0.2073
Human	0.9167	0.9167	0.9167	0.9167	0.8958	0.9167
Zero-Shot						
Direct Prompt	0.688	0.438	0.562	0.375	0.281	0.562
Definition Prompt	0.719	0.719	0.781	0.562	0.531	0.740
Few-Shot						
1 Shot + SE + Reason	0.781	0.688	**0.844**	0.562	0.562	0.771
3 Shot + SE + Reason	0.812	0.656	**0.844**	0.562	0.531	0.771
5 Shot + SE + Reason	0.719	0.719	**0.844**	0.562	0.531	0.760
1 Shot + SE + GPTR	0.562	0.531	0.406	0.500	0.250	0.500
3 Shot + SE + GPTR	0.781	0.625	0.812	0.562	0.500	0.740
5 Shot + SE + GPTR	0.750	0.719	**0.844**	0.625	0.562	0.771
1 Shot + DE	0.844	0.688	0.719	0.656	0.562	0.750
3 Shot + DE	0.844	0.719	0.781	0.656	0.625	0.781
5 Shot + DE	**0.875**	0.781	0.812	**0.750**	**0.656**	**0.823**
1 Shot + DE + Reason	0.844	0.688	0.812	0.656	0.625	0.781
3 Shot + DE + Reason	0.750	**0.812**	0.750	0.656	0.594	0.771
5 Shot + DE + Reason	0.750	**0.812**	0.812	0.688	0.594	0.792

Employing the zero-shot approach of providing the model with the definition of the sentence or word puzzle (Definition Prompting) yields superior performance across both subtasks compared to simply prompting it to indicate the correct option (Direct Prompting). The significant improvement contributing factor to the overall score is evident in the significant improvements observed in the context reconstruction scores(Con).

Based on the results shown in Figure 1 and Table 2, it appears that incorporating GPT-4's reasoning alongside static examples (SE) proved to be the most effective strategy for tackling Sentence Puzzles. The tested strategies also demonstrated improved outcomes when more examples were employed, emphasizing the crucial role of using more in-context examples. However, it is noteworthy that adding self-generated reasoning to the few-shot strategy with dynamic examples did not yield a commensurate improvement; instead, it resulted in a trade-off. While it enhanced the scores for semantically related questions, it came at the expense of the performance on other question types.

Contrary to expectations, the findings in Figure 2 and Table 3 reveal that static examples and reasoning work just as well and even better than dynamic examples and reasoning in the few-shot learning context for the Word Puzzle task. Specifically, incorporating self-generated reasoning alongside dynamic examples led to no significant advancements, indicating that the presumed benefits of dynamic examples did not materialize as expected.

6 Conclusion

In our work, our extensive experimentation demonstrates that the Gemini Pro 1 Model can perform lateral thinking tasks and demonstrate significant improvements in both Sentence Puzzle and Word Puzzle subtasks by employing static and dynamic example selection coupled with self-generated reasoning strategies. We achieved notable enhancements over the baseline models and observed minor improvements as we increased the number of examples used for prompting. While our approach demonstrates notable progress, it still falls short of the performance of human annotators, indicating that further research and development are necessary to bridge this gap.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Sandra Dingli. 2008. Thinking outside the box: Edward de bono's lateral thinking. In *The Routledge companion to creativity*, pages 338–350. Routledge.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, and Hai-Tao Zheng. 2023. Lateval: An interactive llms evaluation benchmark with incomplete information from lateral thinking puzzles.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023a. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023b. Brainteaser: Lateral thinking puzzles for large language models.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13507–13515.

Jorge Martinez-Gil, Bernhard Freudenthaler, and A Min Tjoa. 2019. *Multiple Choice Question Answering in the Legal Domain Using Reinforced Co-occurrence*, pages 138–148.

Zijie Meng, Yan Zhang, Zhaopeng Feng, Yang Feng, Gaoang Wang, Joey Tianyi Zhou, Jian Wu, and Zuozhu Liu. 2024. Divide and conquer for large language models reasoning.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6:245–255.

Yuanzhen Xie, Tao Xie, Mingxiong Lin, WenTao Wei, Chenglin Li, Beibei Kong, Lei Chen, Chengxiang Zhuo, Bo Hu, and Zang Li. 2023. Olagpt: Empowering llms with human-like problem-solving abilities.

A Zero-Shot Prompt Template

Here, we provide the modified zero-shot prompt templates for the sentence and word puzzles.

A.1 Sentence Puzzle

```
Welcome to the sentence-play puzzle challenge! You are presented with a question based on a sentence-play puzzle. It means that the question is a sentence-type brain teaser where the puzzle-defying commonsense is centered on sentence snippets. Remember to pay attention to the details mentioned and indicate the option number you believe is correct for the question:
Question: {question}
Choices:{choices}
```

A.2 Word Puzzle

```
Welcome to the word-play puzzle challenge! You are presented with a question based on a word-play puzzle. It means that the question is a brain teaser where the answer violates the default meaning of the word and focuses on the letter composition of the target question. Remember to pay attention to the details mentioned and indicate the option number you believe is correct for the question:
Question: {question}
Choices:{choices}
```

B Few-Shot Prompt Template

We provide the 2-shot prompt templates for both puzzles here as an example of the few-shot prompting approach.

B.1 Sentence Puzzle

```
Welcome to the sentence-play puzzle challenge! Here, you will be presented with a question based on a sentence-play puzzle. It means that the question is a sentence-type brain teaser where the puzzle-defying commonsense is centred on sentence snippets.
We have given you two examples below to help you understand the puzzle challenge better.
Example 1:
Question: {question}
Choices:{choices}
Correct Option: {correct choice}
Example 2:
Question: {question}
Choices:{choices}
Correct Option: {correct choice}
Now, we shall be giving you the puzzle you need to solve. Remember to pay attention to the details mentioned and indicate the option number you believe is correct for the question:
Question: {question}
Choices:{choices}
```

B.2 Word Puzzle

```
Welcome to the word-play puzzle challenge! Here, you will be presented with a question based on a word-play
```

puzzle. It means that the question is a brain teaser where the answer violates the default meaning of the word and focuses on the letter composition of the target question.
We have given you two examples below to help you understand the puzzle challenge better.
Example 1:
Question: {question}
Choices:{choices}
Correct Option: {correct choice}
Example 2:
Question: {question}
Choices:{choices}
Correct Option: {correct choice}
Now, we shall be giving you the puzzle you need to solve. Remember to pay attention to the details mentioned and indicate the option number you believe is correct for the question:
Question: {question}
Choices:{choices}

C Prompt Template for Self-Generated Reasoning

We provide the self-generated reasoning prompt template for the sentence and word puzzles.

C.1 Sentence Puzzle

CONTEXT
We are presented with a question based on a sentence-play puzzle. It means that the question is a sentence-type brain teaser where the puzzle-defying commonsense is centered on sentence snippets.
OBJECTIVE
We have provided you below with the question, the answer choices and the correct option number and choice. We have also provided you with what the distractor choice was. This distractor was aimed to throw you off the correct answer. You need to provide the reasoning for why the option is correct.
Question: {question}
Choices:{choices}
Correct Option: Option {option number} : {relevant content}
Distractor Choice: {distractor option content}
RESPONSE
Provide the reasoning for why the given correct option is correct and what should be taken care of so that the distractor choice is not chosen.
REASONING

C.2 Word Puzzle

CONTEXT
We are presented with a question based on a word-play puzzle. It means that the question is a brain teaser where the answer violates the default meaning of the word and focuses on the letter composition of the target question.
OBJECTIVE
We have provided you below with the question, the answer choices and the correct option number and choice. We have also provided you with what the distractor choice was. This distractor was aimed to throw you off the correct answer. You need to provide the reasoning for why the option is correct.
Question: {question}
Choices:{choices}
Correct Option: Option {option number} : {relevant content}
Distractor Choice: {distractor option content}
RESPONSE
Provide the reasoning for why the given correct option is correct and what should be taken care of so that the distractor choice is not chosen.
REASONING

D Examples of Self-Generated Reasoning

We have provided examples of the reasoning generated by the Gemini Pro 1 and GPT-4 Models over the training data.

D.1 Reasoning using Gemini Pro 1

D.1.1 Example 1

Question: There was once a bad wizard. He abducted three women from their homes and transformed them into identical rose plants. They were placed in his garden. One of the women pleaded with the magician to allow her to see her husband and kids. He concurred. He took the woman to her

home at night. He arrived early in the morning and drove her home. The spouse made the decision to go get her one day. He entered the wizard's garden as a result. He kept glancing at the three identical rose bushes, trying to determine which one might be his wife. He took his wife home as soon as he realized the solution. How did he recognize his wife among the rosebushes?
Choices:
Option 1: The husband knew which rose bush was his wife because it was the only one that still had her distinctive scent.
Option 2: The wizard brought the rosebush to her home at night and returned her to the garden in the morning. Therefore, she was the only plant without dew.
Option 3: The husband knew which rose bush was his wife because it was the only one that leaned towards him as he approached.
Option 4: None of above.
Correct Option: 2
Reason for Correct Option: Reasoning for the Correct Option:
Option 2: "The wizard brought the rosebush to her home at night and returned her to the garden in the morning. Therefore, she was the only plant without dew." is correct because:
- The passage states that the wizard took the woman to her home at night and returned her to the garden in the morning.
- This means that the rosebush that was his wife would be the only one without dew on it in the morning, as it had been indoors all night.
- The other rosebushes would have dew on them because they had been outside all night.

Distractor Choice Analysis:
Option 1: "The husband knew which rose bush was his wife because it was the only one that still had her distinctive scent." is a distractor choice because:
- The passage does not mention anything about the women retaining their distinctive scents after being transformed into rosebushes.
- Therefore, it is not possible for the husband to identify his wife by her scent.

Conclusion:
Therefore, the correct option is Option 2: "The wizard brought the rosebush to her home at night and returned her to the garden in the morning. Therefore, she was the only plant without dew." because it is the only option that is supported by the information in the passage.

D.1.2 Example 2

Question: What is the most shocking city?
Choices:
Option 1: Urban city.
Option 2: Inner city.
Option 3: Electricity.
Option 4: None of above.
Correct Option: 3
Reason for Correct Option: Reasoning:
The question is "What is the most shocking city?". The correct answer is "Electricity". Electricity is a form of energy that can cause shocks. A city is a large urban area. Therefore, the most shocking city is Electricity.
Distractor Choice:
The distractor choice is "Inner city". Inner city is a part of a city that is typically characterized by poverty and crime. Inner city is not a form of energy that can cause shocks. Therefore, inner city is not the most shocking city.
To avoid choosing the distractor choice, one should:
- Pay attention to the question and the answer choices.
- Understand the meaning of the words in the question and the answer choices.
- Use logic and reasoning to eliminate the incorrect answer choices.

D.2 Reasoning using GPT-4

D.2.1 Example 1

Question: An electric train is going south at 98 mph. The wind is blowing northeast. Which direction is the smoke blowing?
Choices:
Option 1: Northeast.
Option 2: West.
Option 3: South.
Option 4: None of above.
Correct Option: 4

Reason for Correct Option: To solve this sentence-play puzzle, let's analyze the key elements of the question:
Electric Train: This is a crucial piece of information. Electric trains do not produce smoke as they do not burn fuel in the same way that steam or diesel trains do.
Direction of Train: The train is going south. This would only be relevant if we were dealing with a train that produces smoke.
Wind Direction: The wind is blowing northeast. Normally, this would affect the direction of the smoke if the train produced any.
Given these points, the key detail here is that the train is electric. Therefore, the direction of the smoke is a trick question because there would be no smoke produced by an electric train.

D.2.2 Example 2

Question: What kind of lamp emits no light?
Choices:
Option 1: Oil lamp.
Option 2: LED lamp.
Option 3: Clamp.
Option 4: None of above.
Correct Option: 3
Reason for the Correct Option: This puzzle requires us to think beyond the literal uses of the words provided, focusing on the play on words involved in the question and the choices given. The question asks, "What kind of lamp emits no light?" Here are the steps to analyze the choices:
1. Oil lamp. An oil lamp is designed to emit light using oil as fuel. Therefore, it does not fit the criteria as it indeed emits light.
2. LED lamp. An LED lamp uses light-emitting diodes to produce light. Like the oil lamp, it is designed to emit light, so it also does not fit the criteria.
3. Clamp. This option is a play on words. While "clamp" contains the word "lamp," it is not a type of lamp at all; instead, it's a tool used for holding objects tightly together. Since it's not a device designed to emit light, it technically "emits no light."
4. None of above. This option would be correct if none of the first three choices were accurate. However, based on the analysis, there is an option that meets the criteria of emitting no light in the context of the puzzle.
Given the play on words and focusing on the criteria of emitting no light, the correct option is:
3. Clamp. This is because it's the only choice among the options that, despite containing "lamp" in its spelling, does not function as a light-emitting device.

uTeBC-NLP at SemEval-2024 Task 9: Can LLMs be Lateral Thinkers?

Pouya Sadeghi*[1], Amirhossein Abaskohi*[2], and Yadollah Yaghoobzadeh[1,3]

[1]Department of Electrical and Computer Engineering, University of Tehran
[2]Department of Computer Science, University of British Columbia
[3]Tehran Institute for Advanced Studies, Khatam University, Tehran, Iran
[1]{pouya.sadeghi,y.yaghoobzadeh}@ut.ac.ir
[2]aabaskoh@student.ubc.ca

Abstract

Inspired by human cognition, Jiang et al. (2023c) create a benchmark for assessing LLMs' lateral thinking—thinking outside the box. Building upon this benchmark, we investigate how different prompting methods enhance LLMs' performance on this task to reveal their inherent power for outside-the-box thinking ability. Through participating in SemEval-2024, task 9, Sentence Puzzle sub-task, we explore prompt engineering methods: chain of thoughts (CoT) and direct prompting, enhancing with informative descriptions, and employing contextualizing prompts using a retrieval augmented generation (RAG) pipeline. Our experiments involve three LLMs including GPT-3.5, GPT-4, and Zephyr-7B-β. We generate a dataset of thinking paths between riddles and options using GPT-4, validated by humans for quality. Findings indicate that compressed informative prompts enhance performance. Dynamic in-context learning enhances model performance significantly. Furthermore, fine-tuning Zephyr on our dataset enhances performance across other commonsense datasets, underscoring the value of innovative thinking.[1]

1 Introduction

Human cognition provides the foundational framework for understanding large language model (LLM) development, incorporating two critical thinking modes: vertical and lateral (Waks, 1997). Vertical thinking, synonymous with logical reasoning, relies on structured analysis and established principles. Conversely, lateral thinking, known for its creativity, challenges conventions and fosters innovative perspectives, enriching language processing capabilities. Recognizing and leveraging the synergy between vertical and lateral thinking are essential in maximizing LLMs' cognitive potential.

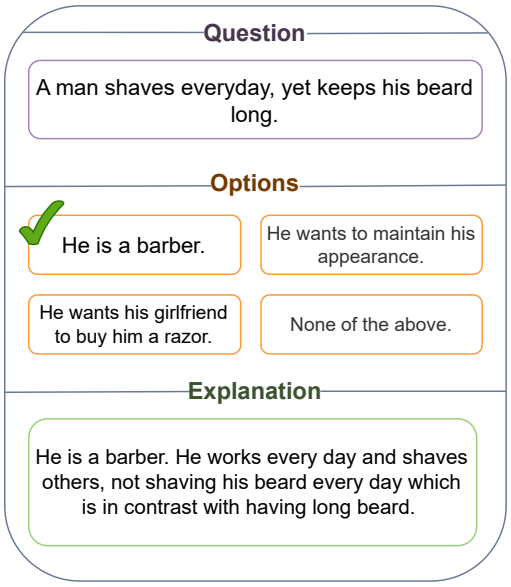


Figure 1: A sample from the sentence puzzle sub-task with an explanation of how this puzzle deprecates default commonsense.

Integrating both strategies facilitates adaptability and ingenuity in addressing linguistic challenges.

Despite LLMs' success and the abundance of reasoning benchmarks (Ho et al., 2023; Abaskohi et al., 2023; Yasunaga et al., 2024), understanding their reasoning remains incomplete. Many benchmarks prioritize vertical over lateral thinking (Waks, 1997), inherent in LLMs' pre-training data. Jiang et al. (2023c) introduces a challenging dataset, yet thorough analyses of prompting methods are lacking.

Building on previous research examining the impact of prompts on LLMs' performance (Webson and Pavlick, 2022), our study aims to validate the genuine lateral understanding capability of LLMs. We participated in SemEval-2024, shared task 9, utilizing various prompts to assess LLMs' lateral thinking abilities in the BrainTeaser multiple-choice QA task (Jiang et al., 2023c, 2024). The task focuses on the Sentence Puzzle (see Fig-

*Equal Contribution
[1]Our codes and data are publicly available at: https://github.com/Ipouyall/Can-LLMs-be-Lateral-Thinkers

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1767–1778
June 20-21, 2024

ure 1) and Word Puzzle sub-tasks[2], challenging common sense associations.

Our team, **uTeBC-NLP**, employs three different methods to evaluate LLMs' lateral thinking ability: (I) chain of thoughts (CoT)-based strategies, (II) enhancing prompts with a detailed task description and prompt compression, and (III) in-context learning ability, using retrieval-augmented generation (RAG) to select dynamic samples. We conducted these experiments on three LLMs: GPT-3.5[3], GPT-4[4], and Zephyr-7B-β (Tunstall et al., 2023), a fine-tuned version of Mistral-7B (Jiang et al., 2023a) trained on a combination of publicly available, synthetic datasets using direct preference optimization (DPO) (Rafailov et al., 2024).

Our contributions include: (I) exploring the impact of incorporating task information on lateral thinking, (II) developing a thesis-based approach wherein we delineate a path between each question-option pair separately, utilizing this thesis as contextual information in subsequent runs—termed as *external-CoT*, (III) leveraging RAG for generating few-shot examples to assess the efficacy of dynamic few-shot inference (IV) employing the generated thesis context to fine-tune Zephyr-7B-β and evaluating its impact on the model's comprehension of commonsense datasets.

In summary, our findings reveal that not all LLMs possess lateral thinking capabilities, with the ability more prominent in models with a greater number of parameters and exposure to extensive data. Proper prompting and introducing unconventional patterns would enhance this capability, by moving beyond conventional linear thinking. Models tend to prefer brief and informative prompts over lengthier alternatives. Notably, we excelled in the Sentence Puzzle sub-task, achieving a remarkable score of 0.975 in solving sentence puzzles, surpassing the baseline of 0.608, and securing the second-highest score.

2 Background

Chain of Thoughts Prompting. In-context zero-shot and few-shot learning play crucial roles in the success of LLMs. To enhance LLM performance across various tasks, including reasoning tasks, as proposed by Wei et al. (2022), we employ the CoT methodology. Replicating Wei et al. (2022)'s setup with sample shots and ensuring their quality can be challenging, so, to ensure fairness and rely on LLMs' knowledge, we adopt Kojima et al. (2022)'s zero-shot-CoT approach and referred to as Simple-Internal-CoT. In our simple-Internal-CoT experiments, we allow LLMs to autonomously handle the problem, thereby separating their performance from the quality of the provided samples. We introduced other variations for CoT, (I) Specifiec-Internal-CoT, in which we explain thinking steps to the LLM, and (II) External-CoT, in which we externally help the LLM to follow the steps by asking it to do only one step in each inference, and preserve results use in next steps' prompts.

Enhanced Prompting Strategies. Prompt strategies, particularly CoT, prove effective in enhancing LLMs' performance across tasks. However, our experiments demonstrate that CoT may not consistently yield better results. Another approach is boosting the model's performance, using informative prompts, inspired from Fernando et al. (2023). We developed a detailed prompt to familiarize LLMs with diverse riddle approaches in a simple and informative manner, aiming to prevent hasty answers and help them to know how should be faced with riddles. Acknowledging the lengthiness of our detailed prompt and to prevent this factor from limiting LLMs' performance, we created a highly compressed version following Jiang et al. (2023b), retaining essential details to improve LLMs' performance while minimizing prompt length.

In-context Learning. In-context learning proved to be one of the most powerful methods to enhance LLMs' performance (Brown et al., 2020). Few-shot prompting with static samples is examined by Jiang et al. (2023c) and shown not to work well. We Employed a RAG pipeline to dynamically select samples from the training split of the dataset. We focused on a three-shot manner and examined different ways of using our RAG pipeline to achieve the best performance. We also explore whether we need to explicitly mention the relation between the question and its answer or whether it can be inferred that effectively.

3 Methodology

This section begins with an overview of the dataset, followed by a detailed exposition of our methodology. We will start by elucidating the task information context. Subsequently, we will explore

[2]We just participated in the Sentence Puzzle sub-task.
[3]We used the gpt-3.5-turbo-0125 version.
[4]We used the gpt-4-0613 version.

different variations of CoT before concluding with an explanation of our RAG methods.

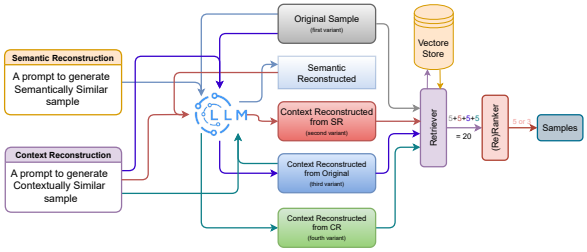


Figure 2: An illustration of our rag-fusion setup. Using an LLM, we generate four variations of the original sample to identify similar ones in the dataset, then rank them to find the closest matches. See appendix D for more details and used prompts.

3.1 Dataset

BrainTeaser. The BrainTeaser dataset (Jiang et al., 2023c) is a multiple-choice question-answering task, designed to evaluate a model's capability for lateral thinking and its ability to challenge default commonsense associations. Created to address the gap in the NLP community's attention towards tasks requiring implicit and intricate reasoning, the dataset relies on human-like commonsense mechanisms. The authors devised a three-step approach to create the first lateral thinking benchmark, involving data collection, distractor generation, and making adversarial examples. They produced 1,100 puzzles with detailed annotations. Assessing models' lateral reasoning consistency, they enhanced BrainTeaser questions with semantic and contextual adjustments. Experiments with top-notch language models showed a significant performance difference from humans, particularly evident across adversarial formats, which aim to avoid cheating in scores by memorizing or previously seen examples. The dataset includes 627 samples for sentence puzzles [5] and 492 samples for word puzzles [6]. In the case of sentence puzzles utilized in our experiments, the average number of tokens in questions is 34.88, with an average of 9.11 tokens in the answers. Our experiments are focused on the Sentence Puzzle sub-task and report our results on test split (post-evaluation phase).

Additional Datasets. We generated a dataset based on BrainTeaser's train data that contains a thinking path between a riddle and each of its options, prompting GPT-4 and revised by authors to avoid any bias. Then fine-tuned Zephyr on this dataset. As explored by Jiang et al. (2023c), Fine-tuning models on vertical thinking and traditional commonsense datasets can't improve performance on BrainTeaser and the model's lateral thinking ability. We aim to evaluate whether the fine-tuned model demonstrates an improvement in general commonsense knowledge and examine how lateral thinking ability could affect the model's performance. We utilized SWAG (Zellers et al., 2018) and CommonsenseQA (Talmor et al., 2019) for this purpose, as they don't need lateral thinking.

SWAG is a vast and diverse dataset designed for grounded commonsense inference, comprising over 113,000 sentence-pair completion examples sourced from internet text. Each example presents a context sentence followed by a partial continuation, prompting the selection of the most plausible completion among four choices.

CommonsenseQA (CSQA) serves as a rigorous benchmark for commonsense reasoning, featuring multiple-choice questions requiring an understanding of everyday situations, world facts, and causal relationships. Questions are associated with concepts from a large commonsense knowledge graph, interconnected through various relations, challenging models to engage in complex commonsense reasoning. Our evaluation incorporates 150 random samples from each of these datasets.

3.2 Task Informative Context

One approach to evaluating whether LLMs possess lateral thinking abilities is to prompt them explicitly for such capabilities. A key strategy involves providing hints about the task, signaling to the model that it should engage in unconventional thinking. In pursuit of this, we design three variations for task description: **(I) Simple**, which doesn't provide any special detail and serves as a base to provide evidence of how description could affect the model's performance, **(II) Detailed**, which would provide detailed information for the task and introduce common tricks to the LLM, and **(III) Compressed**, which is generated from the detailed variation and it just point out instead of detailed explanation. See Appendix A for more details.

3.3 Thinking Strategy

Many different thinking styles are recommended to enhance LLMs' ability to better perform on difficult tasks. CoT prompting has emerged as a potent

[5] Train: 507, Validation: 120, Test: 120
[6] Train: 396, Validation: 96, Test: 96

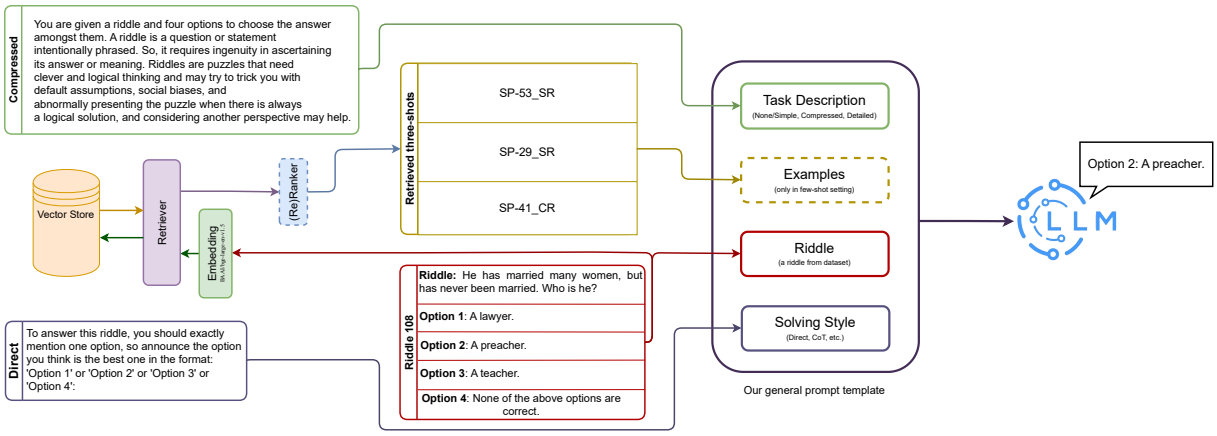


Figure 3: An overview of our approaches in solving the BrainTeaser riddles. In this setup, we have a **direct** prompt that asks the model to find the appropriate answer. To provide more information to the model, we can offer some task explanation, with the **compressed** version depicted in this figure. Finally, we utilize our RAG setup to provide the model with in-context examples. In some experiments, we also include the **theses** for each question-option pair in the prompt, serving as an unbiased link between the question and the option.

strategy for enhancing the LLMs' performance, particularly in tasks requiring complex reasoning such as arithmetic and commonsense reasoning (Zhang et al., 2022; Diao et al., 2023; Zou et al., 2023) and known as a popular choice for these complex tasks. However, the question of how CoT should be implemented remains an open challenge. Moreover, we should be aware that CoT won't achieve the best results in all cases, and use it or not, depends on the task and model.

We consider CoT prompting as two main approaches: **(I) Internal CoT** and **(II) External CoT**. Internal CoT involves guiding the model through step-by-step thinking or incrementally posing questions to facilitate analytical consideration of each option. Our exploration of internal CoT encompasses two types: (I) Simple, and (II) Specified. In Simple Internal CoT, the model is prompted to think step-by-step without explicit specification of each intermediate step. Specified Internal CoT provides the model with explicitly outlined steps to follow in reaching its answer. Conversely, in External CoT, similar to specified-internal-CoT, we defined steps that the model should pass to reach the final answer, but instead of letting the model control the process, we prompt it to do one step in each inference and use the model's response to generate next prompts till we reach to the final answer. Our suggested intermediate reasoning steps, "*find a path between the question and each answer option and then select the most logical one*," are independent for each question-option pair, and referred to as "*thesis*". Then we would use them as context for each option of the riddle and prompt model to solve the riddle regarding provided contexts. In Section 4.1 and Figure 4a, various CoT methods along with direct prompting, are compared.

3.4 In-context Learning

In this method, we let the model learn the task, using sample(s), known as few-shot prompting. In our few-shot experiments, we individually utilized three samples per question (Figure 3). We observed that employing static samples, as traditionally done in few-shot prompts (Brown et al., 2020), did not yield a significant performance boost, supporting few-shot results examined by Jiang et al. (2023c). To overcome this limitation, we developed a RAG pipeline to select shots dynamically based on each question. Our experiments involved three RAG methods: Ordinary RAG, Ranked RAG (RAG+ReRanker), and RAG-Fusion. Our experiments are all the same for these approaches and samples are selected from train split. Our shot instances are available with three entities: question, ground-truth answer, and explanation. Our explanations are sampled using GPT4, and they are a logical thinking path from question to answer. We also generated another variant for explanation named Summarized, which compressed long explanations. See Appendix D for more detail.

Ordinary RAG. Within the ordinary RAG approach, we employed the established RAG framework (Lewis et al., 2020) to generate contextualized representations for the question. The RAG model retrieved relevant passages from a knowl-

edge source, supplying contextual information crucial to the model's decision-making process.

Ranked RAG. The Ranked RAG approach (Lewis et al., 2020) entailed enhancing the RAG framework by incorporating a reranking mechanism. In this variant, retrieved passages would be reranked based on their relevance to the given question, prioritizing those deemed most relevant. This integration is aimed to enhance the quality and relevance of contextual information provided to the model and GPT-4's contexts had the most positive impact[7] on this approach.

RAG Fusion. The RAG-Fusion method (Rackauckas, 2024) seamlessly integrates elements from both ordinary RAG and Ranked RAG. In this methodology, we generate three distinct variants derived from the original riddle, which are subsequently input into the RAG pipeline for sample retrieval. After this step, a ranker[8] is employed to prioritize these samples (Figure 2). This multi-step process is meticulously designed to capture diverse contextual nuances and semantic variations, thereby significantly enhancing the overall effectiveness of the RAG-Fusion method. The most important weakness of this approach is that it is time-consuming as we need many LLM inferences that would take a long time.

Benchmarking. We designed a benchmark, in which we used samples from the train split, and retrieved 5 unique samples at the end, to observe each variant's performance on different setups. for each retrieved sample from the same group, including the sample itself, the method would give one point for that sample[9]as shown on 2 ordinary RAG can satisfy our needs in a more sample-efficient manner.

4 Experiments and Results

In this section, we report and discuss our results. Table 1 shows our submission scores during the competition. We first report and discuss our results on different methods (See Appendix C for full results). Next, we will discuss the effect of fine-tuning on our lateral thinking dataset.

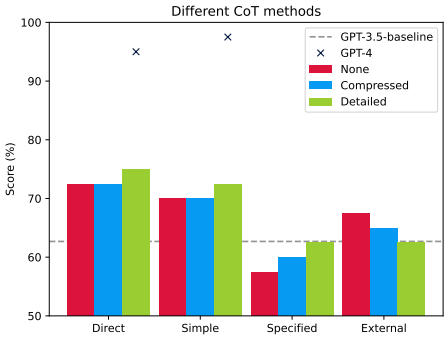
[7]Help this method to provide more helpful samples for our purpose.

[8]The same reranker used as Ranked RAG.

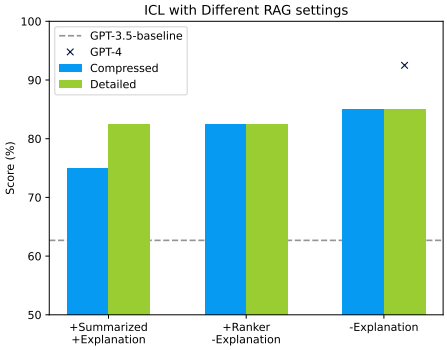
[9]Max points for one sample: 3, as we have three variations in each group.

Metric	Score (%)
Ori	97.5
Sem	87.5
Con	82.5
Ori&Sem	85.0
OriSemCon	75.0
Overall	89.2

Table 1: Our Final Submission for Sentence Puzzle subtask. This submission was made by GPT-4 in Simple-Internal-CoT and detailed task description setting.



(a) Effect of different thinking (solving) styles on model's performance. Bars belong to GPT-3.5.



(b) Our proposed RAG-pipelines for dynamic in-context learning(Direct Prompting) and its effect on the model's performance. Bars belong to GPT-3.5.

Figure 4: Different prompting approaches and how they affect the model's performance. GPT-3.5-baseline reported by Jiang et al. (2023c).

Used Samples	RAG Type	Hit Rate
20	Ordinary	0.65
	Fusion	**0.767**
	Ranker	0.65
507	Ordinary	**0.753**
	Ranker	0.73

Table 2: Results of RAG's variants on the train split.

4.1 Prompting Methods

In this section, we detail our exploration of various prompting methods, as outlined in Section 3. Figure 4a presents a comparative analysis, revealing that, among different CoT methodologies, simple internal CoT exhibits superior performance. However, it scores lower than direct prompting, without any CoT variation employed. Notably, external CoT outperforms specifying steps in one prompt (specified internal CoT) but falls short compared to simple internal CoT and direct prompting. This is attributed to the impact of prompt length, where longer prompts with similar information weaken performance.

Prompt Length In task descriptions, providing hints consistently aided the model, with the condensed and detailed versions excelling in different scenarios. Our hypothesis, supported by the results, posits that both prompt and cognitive pathway length significantly influence performance. Extensive factors lead the model to favor concise yet informative prompts, as evidenced by the superior performance with the compressed descriptions.

In-context Learning As explained in Section 3, we focused on repeating experiments with Ordinary RAG and RAG+(Re)Ranker in three-shot format, leveraging three entities: question, ground-truth answer, and explanation.

Our explanations, sampled using GPT-4, represent a logical thinking path from question to answer. Additionally, we introduced a summarized variant, generated through Cohere's summarize API [10] [11] for explanations exceeding 250 words. As depicted in Figure 4b, using (Re)Ranker does not significantly enhance performance. Increasing task description details improves performance. Interestingly, we observed that excluding explanation and letting the model infer the thinking path between riddles and their answer will boost LLMs' performance, proving LLMs' ability to extract relations and thinking paths independently.

Examining three semantically similar questions in a three-shot format, the LLMs' performance converges to a certain score, independent of the task description. The optimal performance is achieved when excluding explanations and using a simple RAG pipeline without using (re)ranker. See Appendix D for more details on settings.

[10] https://cohere.com/summarize

[11] Settings: length="short", extractiveness="high"

4.2 Lateral Thinking Tuning

Fine-tuning is a core strategy for refining model performance in specific tasks. However, as noted in Jiang et al. (2023c), fine-tuning on other commonsense datasets may not guarantee performance improvements; potentially, it may even lead to a decline in the overall model's performance. This experiment focuses on fine-tuning the model using the dataset generated by GPT-4 and revised by authors, where the model discerns paths between each riddle and its options. The goal is to assess the impact of lateral thinking on model performance by evaluating it on other commonsense datasets. We tried to keep the dataset unbiased, enabling LLM to learn lateral thinking ability.

Model	Dataset	Tuned	Accuracy
Zephyr-7B-β	SWAG	No	38
		Yes	**46**
	CSQA	No	31.33
		Yes	**36**

Table 3: Fine-tuning experiments, observing model's performance improvement in commonsense.

Table 3 showcases that fine-tuning with a lateral thinking approach significantly enhances the model's performance on other commonsense datasets. This result challenges the conventional belief that linear thinking might constrain the model in scenarios requiring unconventional solutions. Adopting an out-of-the-box thinking approach proves beneficial, emphasizing the importance of lateral thinking across diverse contexts.

5 Conclusion

In conclusion, our study emphasizes the crucial role of prompting methods in augmenting the lateral thinking capabilities of LLMs. Through diverse CoT-based strategies, prompt refinements, and RAG techniques for in-context learning, we showcase the efficacy of well-structured prompts and thinking styles in elevating LLM performance. Additionally, fine-tuning models on a lateral thinking dataset proves advantageous, leading to improved performance on various commonsense tasks. This underscores the significance of integrating out-of-the-box thinking in model training, opening promising avenues for future research to enhance LLMs' reasoning abilities.

References

Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. 2023. LM-CPPF: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 670–681, Toronto, Canada. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023c. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Zackary Rackauckas. 2024. Rag-fusion: a new take on retrieval-augmented generation. *arXiv preprint arXiv:2402.03367*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6:245–255.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In

Advances in Neural Information Processing Systems, volume 35, pages 24824–24837. Curran Associates, Inc.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. 2024. Large language models as analogical reasoners. In *The Twelfth International Conference on Learning Representations*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Anni Zou, Zhuosheng Zhang, Hai Zhao, and Xiangru Tang. 2023. Meta-cot: Generalizable chain-of-thought prompting in mixed-task scenarios with large language models. *arXiv preprint arXiv:2310.06692*.

A Task description Prompts

During our analysis, we explored various combinations of prompting methodologies. Table A.1 presents our prompts for task description. The initial prompt solely defines what a riddle entails. In the compressed version, we provide general hints, such as avoiding bias. The final prompt includes eleven potential tricks that may occur in the question, along with instructions for the model to evade biases and consider superpower abilities used in the questions. These eleven hints were extracted from other sources.

Prompt
A riddle is a question or statement intentionally phrased so as to require ingenuity in ascertaining its answer or meaning.
A riddle is a question or statement intentionally phrased so as to require ingenuity in ascertaining its answer or meaning. Riddles are puzzles that need clever and logical thinking, and may try to trick you with default assumptions, social biases, and abnormally presenting the puzzle when there are always a logical solution, and considering another perspective may help.
A riddle is a question or statement intentionally phrased so as to require ingenuity in ascertaining its answer or meaning. Different ideas can be used in riddles to trick you: 1. Riddles often employ misdirection, leading you away from the actual solution. 2. They include elements with double meanings, requiring a keen eye for words with dual interpretations. 3. Metaphorical wordplay adds another layer, urging you to decipher figurative language. 4. Look out for exaggeration, as riddles may present overly dramatic details to divert your attention. 5. Common phrases and sayings may hide within the puzzle, demanding familiarity. 6. Associations and irony play a crucial role, introducing unexpected connections. 7. Numerical puzzles can also be part of the mystery, requiring you to decode their significance. 8. Elemental imagery, drawn from nature, might hold key descriptors. 9. Rhyming and sound clues can add a poetic dimension. 10. Avoid sexism and sex cliché, for example, gender bias for jobs, based on their positions or their outcome. 11. Riddle may try to present something impossible or in contradiction with the reality. Just consider alternative perspectives.

Table A.1: Our task description prompts. The first prompt lacks task details, the second is compressed, and the last is detailed, covering all potential tricks.

B Generating Path Between Question and Answer

One of the methods that was used both in external CoT and step-by-step internal CoT was asking the model to generate a thinking path between a question-option pair. To do that, we asked the model to generate a path between question and option, without giving any judgment on the answer and considering every option can be an answer to the question. To prompt we used to was: "`Your task is to generate a descriptive explanation from a question to an answer option. In the following, a question and an option as the answer to the question are provided. The provided option might or not be a correct answer. Write a descriptive explanation in at most one paragraph and 200 words to show the thinking path from the question to the option.`." To avoid hallucination, we tried to limit the model's description to 200 by asking the model to do so. Although it does not work all the time,

it limits the model's generated words. This limitation would be later beneficial as long input prompts could decrease the model's performance.

C Complete experiments and results

In this section, we will show our complete results for the sentence puzzle sub-task. Our scores on different experiments are shown in Table C.1 and each experiment's descriptions are available in Sections 3 and 4.

D RAG extensive experiments

In this section, we discuss our rag experiments in more detail. First, we mention the common setting between different methods, then we will mention each method and explain its specific experimental setup in more detail.

Commonly, we used the Chroma [12] as our vector store. We employed "bge-large-en-v1.5"(Xiao et al., 2023) as our embedding. Our final samples are chosen as the first three unique[13] samples retrieved from our vector store.

Our initial Explanations are selected from the same dataset used for lateral thinking tuning, using ground-truth answers instead of all options. Also for our summarized variant, for explanations longer than 250 words, we used summarizer (in this case, Cohere's summarize API) to summarize explanations.

Overall, our RAG methods can extract different variations in a group, as seen in Table 2 and Origian and SR variations seem to be closely related, but in some cases, it face problems to related CR variation into two other variations.

Ordinary RAG. We have used it as its ordinary usage. Despite having a close performance to RAG+Ranker, we decided to use this method, reducing the ranker's effect on our performance.

RAG+Ranker In this method, we first use a normal retriever as Ordinary RAG to retrieve 25 samples from our vector store. Then we fed our query and retrieved 25 samples to reranke(ranker)[14] and kept the first 3 samples with the highest scores.

RAG Fusion In this method. We designed two prompts to generate semantically or contextually related (see Table D.1). Using those two prompts, by prompting Zephyr-7B-β, we generate three new variations from the original, and counting the original sample, we feed each variation to the retriever to retrieve 5 samples, which provides 20 samples. Then we would run deduplication to eliminate duplicated samples, which may caused by employing multiple retrieval phases, and rank remained samples using a ranker(re-ranker) and keep the first five samples with the highest score (see Figure D.1). The we just continue with the first three samples as our shots.

[12] https://github.com/chroma-core/chroma

[13] The "unique" term comes meaningful with rag-fusion, as it may retrieve the same samples in each retrieval phase

[14] We used Cohere's reranker: https://txt.cohere.com/rerank/

Model	Thinking Method	In-Context Learning	Task Description	Result
GPT 3.5	Direct	-	None	72.5
			Compressed	72.5
			Detailed	75
	Simple-Internal-CoT	-	None	70
			Compressed	70
			Detailed	72.5
	Specified-Internal-CoT	-	None	57.5
			Compressed	60
			Detailed	62.5
	External-CoT	-	None	67.5
			Compressed	65
			Detailed	62.5
	Simple-Internal-CoT	ES	Compressed	72.5
	Direct	ES	Compressed	75
	Direct	ES	Detailed	82.5
	Simple-Internal-CoT	ER	Compressed	72.5
	Direct	R	Compressed	82.5
	Direct	R	Detailed	82.5
	Direct	ord	None	85
	Direct	ord	Compressed	85
	Direct	ord	Detailed	85
	Simple-Internal-CoT	ord	Compressed	77.5
	Specified-Internal-CoT	ord	Compressed	67.5
GPT 4	Direct	-	Detailed	95
	Simple-Internal-CoT	**-**	**Detailed**	**97.5**
	Direct	ord	Compressed	92.5
Zephyr-7B-β	Direct	-	None	27.5
			Detailed	32.5
	Simple-Internal-CoT	-	Compressed	37.5
			Detailed	15
		ER	Compressed	40
		ES	Compressed	42.5
		ESR	Compressed	35
		ord	Compressed	25
		R	Compressed	22.5

Table C.1: Our complete submission result for the post-evaluation phase on test split. In-context learning means using three shots dynamically selected by our RAG's pipeline, in which: **E**) use Explanation, **S**) use Summarizer, **R**)use Ranker, and **ord**) using ordinary rag without explanation and ranker. Our final submission is underlined.

	Prompt
Semantically Related	Semantic reconstruction involves rephrasing a given text while preserving its original meaning. In this context, you are presented with a riddle. The task is to rephrase the riddle without altering the correct answer. Perform a semantic reconstruction of the following riddle. ORG Riddle: "Four men were in a boat on the lake. The boat turns over, and all four men sink to the bottom of the lake, yet not a single man got wet! Why?" SR Riddle: "A boat on the lake included four men. All four men on the boat sink to the bottom of the lake when it flips over. However, not a single man gets wet! Why?" ORG Riddle: "A plane crashed, and every single person on board this flight was killed, yet there were survivors. Explain how?" SR Riddle: "Despite the fact that the entire flight was lost in a plane crash and each single person is killed, there were survivors. Describe how?" ORG Riddle: "{riddle}" SR Riddle:
Contextually Related	Context reconstruction involves maintaining the original reasoning path while changing both the question and the answer to describe a new situational context. In this context, you are presented with a riddle. The task is to reconstruct the context of the riddle while keeping the original reasoning intact. Ensure that the reconstructed context maintains the same reasoning path as the original riddle and also it is reasonable. You should change the context and try to avoid it by just replacing some entities and trying to convey the question to another scenario. Perform a context reconstruction of the following riddle. ORG Riddle: "A woman shoots her husband. Then she holds him underwater for over 5 minutes. Finally, she hangs him. But 5 minutes later, they both go out and enjoy a wonderful dinner together. How can this be?" CR Riddle: "A woman shoots publicly at people at a National Park. The park is full of people, but no one gets killed. How is that possible?" ORG Riddle: "There are 3 apples available for 2 fathers and 2 boys to consume. They each receive a single apple. How is it a mathematical possibility?" CR Riddle: "Two mothers and two daughters were asking for new state IDs, but the agent only gave out three forms and instructed them on how to fill them out. Why?" ORG Riddle: "{riddle}" CR Riddle:

Table D.1: Prompts that are used to generate SR and CR-related samples for the RAG-Fusion method.

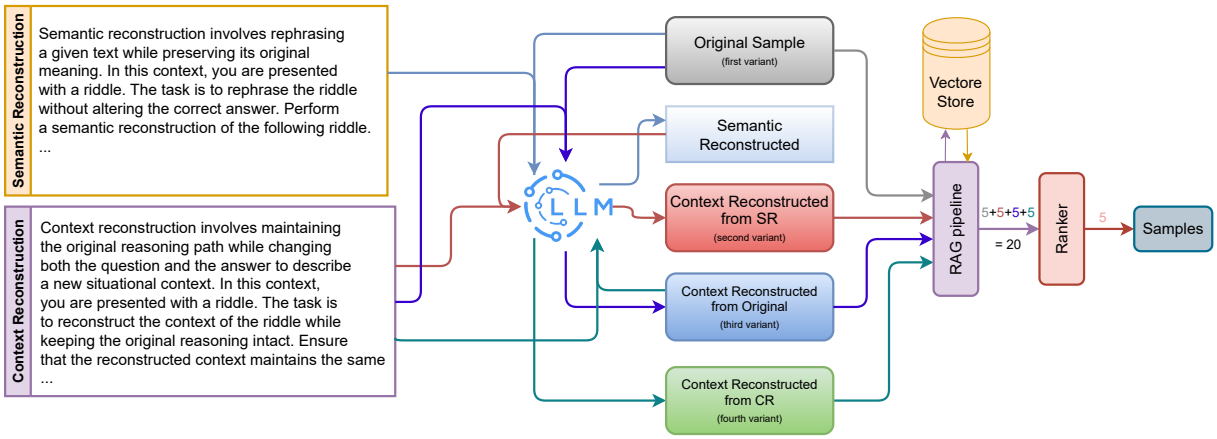


Figure D.1: RAG Fusion. The four used variants include: (I) The original riddle, (II) Context reconstruction obtained from semantically reconstructed samples originating from the original riddle, (III) Context reconstruction derived from the original riddle, (IV) Context reconstructed from step 3, then we retrieve similar samples for each variant. In the end, we feed retrieved documents to a ranker to filter them based on similarity and usefulness.

IITK at SemEval-2024 Task 4: Hierarchical Embeddings for Detection of Persuasion Techniques in Memes

Shreenaga Chikoti **Shrey Mehta** **Ashutosh Modi**
Indian Institute of Technology Kanpur (IIT Kanpur)
chikoti20@iitk.ac.in
ashutoshm@cse.iitk.ac.in

Abstract

Memes are one of the most popular types of content used in an online disinformation campaign. They are primarily effective on social media platforms since they can easily reach many users. Memes in a disinformation campaign achieve their goal of influencing the users through several rhetorical and psychological techniques, such as causal oversimplification, name-calling, and smear. The SemEval 2024 Task 4 *Multilingual Detection of Persuasion Technique in Memes* on identifying such techniques in the memes is divided across three sub-tasks: (**1**) Hierarchical multi-label classification using only textual content of the meme, (**2**) Hierarchical multi-label classification using both, textual and visual content of the meme and (**3**) Binary classification of whether the meme contains a persuasion technique or not using it's textual and visual content. This paper proposes an ensemble of Class Definition Prediction (CDP) and hyperbolic embeddings-based approaches for this task. We enhance meme classification accuracy and comprehensiveness by integrating HypEmo's hierarchical label embeddings (Chen et al., 2023) and a multi-task learning framework for emotion prediction. We achieve a hierarchical F1-score of 0.60, 0.67, and 0.48 on the respective sub-tasks.

1 Introduction

Memes are popular among people of all age groups today through different social media platforms (Keswani et al., 2020; Singh et al., 2020). These memes help people know about the trends around them and can influence their decisions. Memes are one of the popular modes for spreading disinformation among people (examples in Figure 1), as studies have suggested that people tend to believe what they see frequently in such memes spread over the internet (Moravec et al., 2018). As evidenced by research (Shu et al., 2017) during the 2016 US Presidential campaign, nefarious actors, including bots, cyborgs, and trolls, leveraged memes to evoke emotional reactions and propagate misleading narratives (Guo et al., 2020).

Figure 1: Sample set of memes showing the multi-modal setting

In this respect, SemEval-2024 Task 4 (Dimitrov et al., 2024) focuses on predicting the persuasive technique (from the visual and textual content) used in a meme across four different languages: English, Arabic, North Macedonian and Bulgarian. The task is divided into three sub-tasks: (**1**) Hierarchical multi-label classification using only textual content of the meme, (**2**) Hierarchical multi-label classification using both textual and visual content of the meme and (**3**) Binary classification of whether the meme contains a persuasion technique or not using it's textual and visual content. The training data is provided for each sub-task but only in English. Taxonomy of various persuasion techniques (Figure 2) and their respective definitions are provided.

To address sub-task **1**, we employed a dual approach involving definition-based modeling for each class and hierarchical classification using hyperbolic embeddings, as proposed in Chen et al. (2023). Based on hyperbolic embeddings, the

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1779–1787

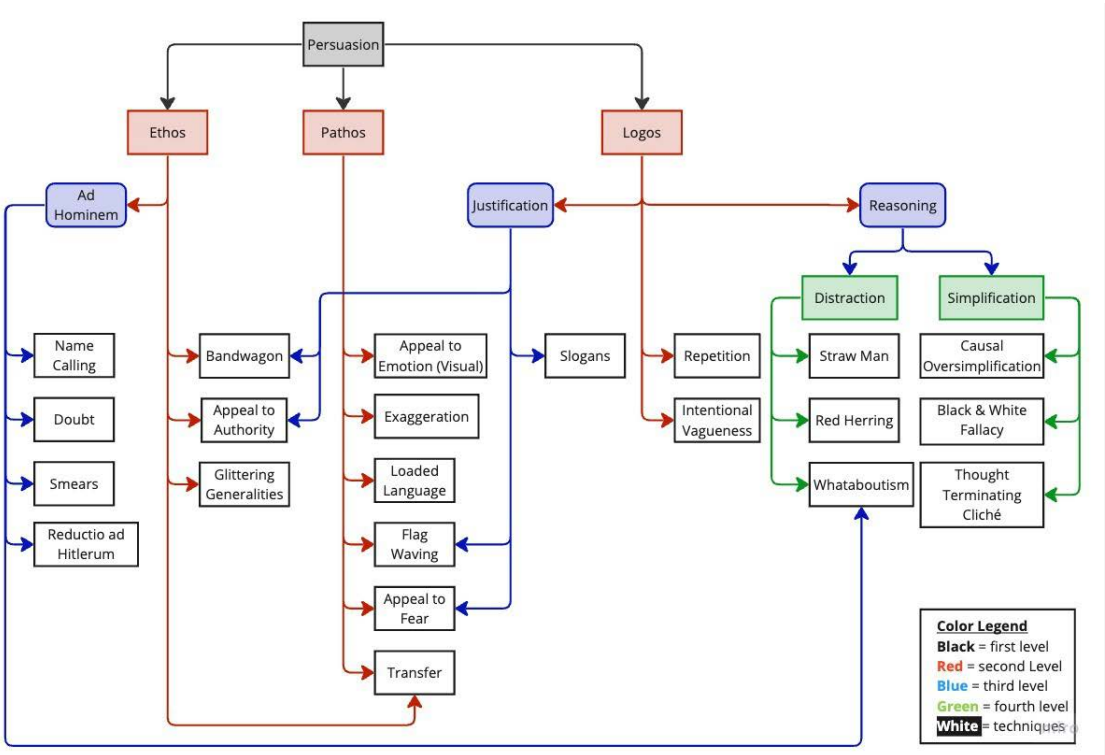


Figure 2: Taxonomy of persuasion techniques for sub-task **2**

method facilitates a nuanced classification of persuasion techniques by leveraging hierarchical structures. The incorporation of definition-based modeling allows for a dataset-agnostic approach, enhancing the precision of classification without reliance on hierarchical structures.

For sub-task **2**, we augmented our methodology by integrating CLIP embeddings (Radford et al., 2021) to capture essential features from memes' textual and visual components. This fusion of textual and visual information enables a more comprehensive analysis of meme content.

In addressing sub-task **3**, we adopted an ensemble approach, leveraging transfer learning from both the DistilBERT (Sanh et al., 2019) and CLIP embeddings (Radford et al., 2021). This ensemble technique enhances the robustness and effectiveness of our classification system by amalgamating insights from both pre-trained models. We release the code via GitHub.[1]

2 Background

The goal of propaganda is to enhance people's mindsets (Singh et al., 2020), especially at the time of elections, where the trends in the media influence the votes of the people (Shu et al., 2017). Propaganda uses psychological and rhetorical techniques to serve its purpose. Such methods include using logical fallacies and appealing to the audience's emotions. Logical fallacies are usually hard to spot since the argumentation, at first sight, might seem correct and objective. However, a careful analysis shows that the conclusion cannot be drawn from the premise without misusing logical rules (Gupta and Sharma, 2021). Another set of techniques uses emotional language to induce the audience to agree with the speaker only based on the emotional bond that is being created, provoking the suspension of any rational analysis of the argumentation (Szabo, 2020).

Corpora development has been instrumental in advancing deception detection methodologies. Rashkin et al. (2017) introduced the TSHP-17 corpus, providing document-level annotation across four classes: trusted, satire, hoax, and propaganda. However, their study on the classification task revealed limitations in the generalizability of n-gram-based approaches. Building on this, Barrón-Cedeno et al. (2019) contributed the QProp corpus, which specifically targeted propaganda detection, employing a binary classification scheme of propaganda versus non-propaganda. Similarly, Habernal

[1] https://github.com/Exploration-Lab/IITK-SemEval-2024-Task-4-Pursuasion-Techniques

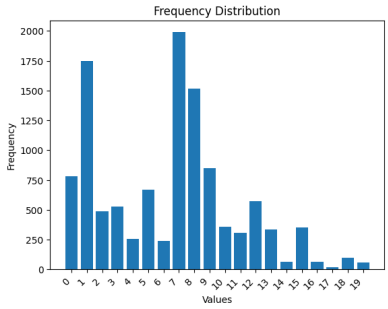


Figure 3: The frequency Distribution of Labels in the training dataset

et al. (2018) developed a corpus annotated with fallacies, including *ad hominem* and *red herring*, directly relevant to propaganda techniques.

BERT-based variants have emerged as promising methodologies for classification tasks in tandem with corpus development. Yoosuf and Yang (2019) proposed a fine-tuning approach post-world-level classification using BERT, while Fadel et al. (2019) presented a pre-trained ensemble model integrating BiLSTM, BERT, and RNN components. Further extending the capabilities of BERT, Costa et al. (2023) advocated for a multilingual setup, employing translation to English before utilizing RoBERTa. Additionally, Teimas and Saias (2023) proposed a hybrid technique combining CNN with DistilBERT for improved detection accuracy.

Exploring multimodal content, Glenski et al. (2019) delved into multilingual multimodal deception detection, mainly focusing on hateful memes. Leveraging visual and textual content, they utilized fine-tuning techniques with state-of-the-art models like ViLBERT and VisualBERT and transfer learning-based approaches (Gupta et al., 2021).

3 Data Description

The competition consisted of two different phases mainly the development phase which we refer to as the development set and for the development phase we were provided the training and validation sets for benchmarking our models

All three sub-tasks have different sets of memes split across training, validation and Develepmont sets as shown in Table 2. We have also plotted the Distribution of the labels across the Figure 3 training data and the Figure 4 validation data.

Our analysis used a dictionary to map various rhetorical techniques to numerical values for plotting. This dictionary is as follows:

Persuasion Technique	**Number mapped to**
Presenting Irrelevant Data (Red Herring)	0
Bandwagon	1
Smears	2
Glittering generalities (Virtue)	3
Causal Oversimplification	4
Whataboutism	5
Loaded Language	6
Exaggeration/Minimisation	7
Repetition	8
Thought-terminating cliché	9
Name calling/Labeling	10
Appeal to authority	11
Black-and-white Fallacy/Dictatorship	12
Obfuscation, Intentional vagueness, Confusion (Straw Man)	13
Reductio ad hitlerum	14
Appeal to fear/prejudice	15
Misrepresentation of Someone's Position (Straw Man)	16
Flag-waving	17
Slogans	18
Doubt	19

Table 1: Dictionary Mapping for different persuasion techniques for Subtask 1

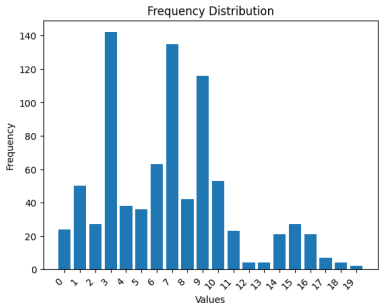


Figure 4: The frequency Distribution of Labels in the validation dataset

4 System overview

The proposed system for all the sub-tasks involves task-specific modifications made to the BERT model and earlier proposed works including CLIP Model (Radford et al., 2021), Class Definition based Emotion Predictions (Singh et al., 2021, 2023) and HypEmo model (Chen et al., 2023) (described below).

4.1 Data Pre-processing

To ensure consistency and standardization, we begin by pre-processing the text. This involves removing newline characters, commas, numerical values, and other special characters. Additionally, the entire text is converted to lowercase. In our approach, we leverage the Development (Dev) and Training sets, focusing solely on samples containing non-zero classes.

Figure 5: The meme sarcastically suggests that individuals who oppose Trump are being unfairly equated with terrorists, highlighting the absurdity of such comparisons. Two persuasion techniques are used: (i) *Loaded Language*, and (ii) *Name calling* that can be inferred from the text and the visual content.

Sub-task	Train Data	Validation Data	Development Data
Sub-task 1	7000	500	1000
Sub-task 2	7000	500	1000
Sub-task 3	1200	300	500

Table 2: Distribution of data across sub-tasks

4.2 Sub-task 1: Hierarchical Multi-label Text Classification

We present a novel approach to meme classification, drawing upon the methodologies of two key frameworks: HypEmo and a multi-task learning model focused on emotion definition modeling.

HypEmo (Chen et al., 2023) utilizes pre-trained label hyperbolic embeddings to capture hierarchical structures effectively, particularly in tree-like formations. Initially, the hidden state of the [CLS] token from the RoBERTa backbone model is projected using a Multi-Layer Perceptron (MLP). Subsequently, an exponential map is applied to project it into hyperbolic space. The distance from pre-trained label embeddings is the weight for the cross-entropy loss function, enhancing the model's sensitivity to label relationships.

To implement the HypEmo architecture, we transform the Directed Acyclic Graph (DAG) (Figure 2) into a tree structure. This involves duplicating children with multiple parents, resulting in distinct embeddings for each label. For example, a sentence with various labels is converted into separate samples, each assigned one label. Utilizing the Poincaré hyperbolic entailment cones model (Ganea et al., 2018) with 100 dimensions, the constructed tree undergoes training, with predictions generated via softmax. Peaks are identified through Z-score analysis associated with each class, with thresholds set accordingly.

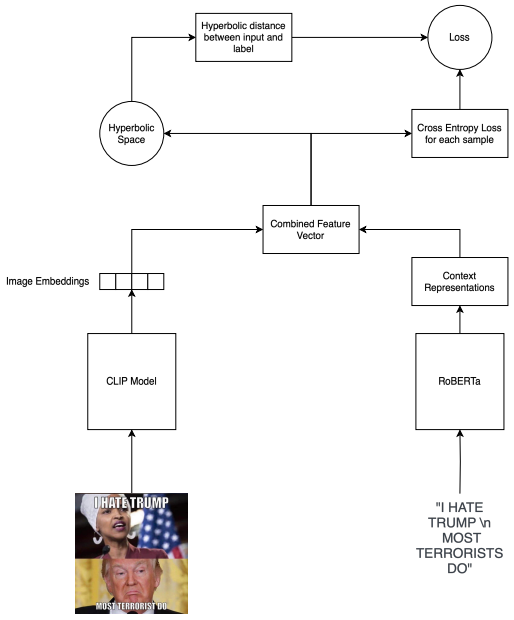


Figure 6: Proposed architecture for sub-task **2**

Singh et al. (2021, 2023) have introduced a complementary approach focusing on emotion prediction through a multi-task learning framework. This model incorporates auxiliary tasks, including masked language modeling (MLM) and class definition prediction, to enhance the understanding of emotional concepts. In our setup, class definitions are merged using a [SEP] token, with the model trained to predict whether the conjoined definition matches the actual definition. Binary cross-entropy loss is employed for this task, along with MLM for fine-tuning the model. Additionally, binary cross-entropy loss is used for each class during training. We utilize class definitions provided by the meme classification competition for the auxiliary task of class-definition prediction.

Finally, we merge the predictions generated by both models (HypEmo, Fine-grained class-definition based model) to compute the final predictions. This integrated approach aims to leverage the strengths of each framework, enhancing the accuracy and comprehensiveness of meme classification outcomes.

4.3 Sub-task 2: Hierarchical Multi-label Text and Image Classification

We model this sub-task by experimenting with using an ensemble of HypEmo (Chen et al., 2023)

and the class definition-based multi-task learning model (Singh et al., 2021, 2023) for the textual content of the meme and using the CLIP model (Radford et al., 2021) embeddings for extracting the relevant features from the visual content of the meme. We construct a similar DAG structure for sub-task 1 and generate the hyperbolic embeddings. The image embeddings obtained from the CLIP model are concatenated with the embeddings generated for the textual contents before sending the combined feature vector for training. Then, the model is trained, and the predictions are generated using the softmax activation function. The Z-score analysis is done on the resulting predictions to make the classification, similar to task 1. An overview of the architecture of the modified HypEmo model is shown in Figure 7.

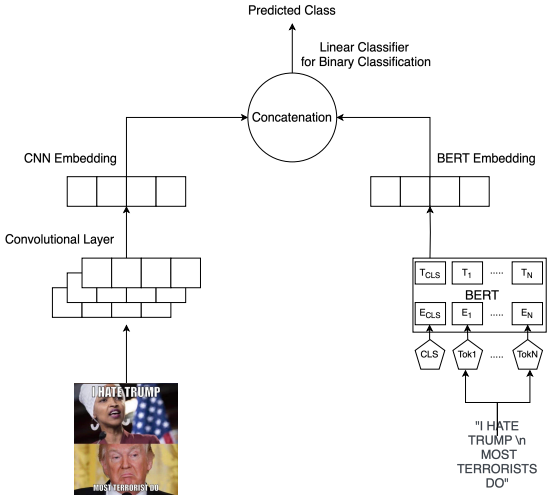


Figure 7: Proposed architecture for sub-task **3**

4.4 Sub-task 3: Binary Text and Image Classification

In this task, we must classify whether a meme contains a persuasion technique based on its textual and visual content. We use the pre-trained $\text{BERT}_{\text{BASE}}$ model (Devlin et al., 2019) and the Convolution Neural Network (CNN) (O'Shea and Nash, 2015) layer to extract the features from the text and image, respectively. We attach a feed-forward $[CLS]$ token embedding along with two linear layers connected by the $sigmoid$ activation function in between, which generates the sentence embeddings corresponding to the textual content in the meme. We use a network of four CNN layers connected through the ReLU activation function, which progressively extracts features from the input image. Max pooling layers are used to downsample the feature maps, increasing robustness to minor variations. The resultant image embeddings are concatenated with the sentence embeddings, and a linear classifier is applied to the combined feature vector with the $sigmoid$ activation function. We use the binary cross-entropy loss function to train the model and tune the hyperparameters on the validation set. An overview of the model architecture is shown with an example in Figure 8.

Since the training data is in a 2:1 ratio for the "persuasive" (positive, labeled as 1) and "not-persuasive" (negative, labeled as 0) class, which leads to an imbalance in the dataset, we use the weighted binary cross entropy loss function as shown below:

$$L(\mathbf{x}, \mathbf{y}) = -\frac{1}{N}\sum_{1}^{N}(w * y_i * log(x_i) + (1-w) * (1-y_i) * log(1-x_i))$$

$$w = \frac{1}{f}(K - f)$$

where N is the batch size, i is the index of the i^{th} batch element, f is the frequency of the positive class, $\mathbf{x}$ is the output of the last $sigmoid$ layer, $\mathbf{y}$ is the vector of the ground truth labels, and K is the total size of the training dataset. Finally, by choosing the one with a higher probability, we use the output probabilities of the final $sigmoid$ layer to predict whether a persuasion technique is present in the meme.

5 Experimental setup

5.1 Implementation Details

We have used the official PyTorch implementation (Paszke et al., 2019) for implementing all the models across sub-tasks. We have used the HypEmo[2] model and the Class Definition Prediction (CDP)[3] model for generating the hyperbolic embeddings and class-definition based features of the textual contents, respectively and the CLIP[4] mainly the 'clip-ViT-B-32' model for generating embeddings for the visual features of the meme. Some portions of the test set have languages other than English for

[2]HypEmo, https://github.com/dinobby/hypemo

[3]CDP, https://github.com/Exploration-Lab/FineGrained-Emotion-Prediciton-Using-Definitions

[4]CLIP, https://github.com/openai/CLIP

testing purposes. Since the models described earlier were trained in English, we translated the non-English data into English language using the implementation of the OPUS-MT model (Tiedemann and Thottingal, 2020) from the HuggingFace[5] library and inference was done on the translated text. We created an ensemble of classes predicted by all the models and took a union of the predicted labels to produce the final predicted set of labels to which the meme belonged.

We have used the data in the same ratio provided in the task to train the models. We combine the train validation dataset for training in each subtask and test it in the four languages.

5.2 Evaluation Metrics

Sub-tasks 1 and 2 depend on a hierarchy, as shown in Figure 2. Hierarchical-F1 (Kiritchenko et al., 2006) is used as the evaluation metric for these two sub-tasks. In these two, the gold label is always a leaf node of the DAG, considering the hierarchy in Figure 2 as a reference. However, any node of the DAG can be a predicted label with:

- If the prediction is a leaf node and it is the correct label, then a full reward is given. For example, *Red Herring* is predicted and is the gold label.
- If the prediction is NOT a leaf node and an ancestor of the correct gold label, then a partial reward is given (the reward depends on the distance between the two nodes). For example, if the gold label is Red Herring and the predicted label is *Distraction* or *Appeal to Logic*.
- If the prediction is not an ancestor node of the correct label, then a null reward is given. For example, if the gold label is *Red Herring* and the predicted label is *Black and White Fallacy* or *Appeal to Emotions.*

Sub-task 3 uses macro-F1 as the evaluation metric for the binary classification task. This ensures equal importance to the "persuasion technique present" and "no persuasion technique" classes, regardless of potential data imbalance.

6 Results

We conducted several experiments across all the sub-tasks, and the detailed information can be seen in Table 3,Table 4,Table 5,Table 8 and Table 9.

[5]OPUS-MT, https://huggingface.co/Helsinki-NLP/opus-mt-bg-en

Technique	Arabic	Bulgarian	North Macedonian
Presenting Irrelevant Data (Red Herring)	0.	0.	0.
Bandwagon	0.	0.	0.
Smears	0.67	0.84	0.90
Glittering generalities (Virtue)	0.29	0.10	0.
Causal Oversimplification	0.	0.	0.
Whataboutism	0.	0.05	0.
Loaded Language	0.41	0.62	0.37
Exaggeration/Minimisation	0.	0.	0.
Repetition	0.50	0.34	0.
Thought-terminating cliché	0.	0.19	0.
Name calling/Labeling	0.44	0.45	0.49
Appeal to authority	0.	0.30	0.31
Black-and-white Fallacy/Dictatorship	0.	0.06	0.
Obfuscation, Intentional vagueness, Confusion (Straw Man)	0.	0.	0.
Reductio ad hitlerum	0.	0.	0.
Appeal to fear/prejudice	0.04	0.21	0.1
Misrepresentation of Someone's Position (Straw Man)	0.	0.	0.
Flag-waving	0.	0.33	0.
Slogans	0.	0.43	0.16
Doubt	0.	0.15	0.11
Transfer	0.	0.48	0.61
Appeal to (Strong) Emotions	0.	0.18	0.09

Table 3: Macro F1 scores for different persuasion classes for the given languages for Subtask 2

Technique	Arabic	Bulgarian	North Macedonian
Presenting Irrelevant Data (Red Herring)	0.	0.	0.
Bandwagon	0.	0.	0.
Smears	0.33	0.18	0.17
Glittering generalities (Virtue)	0.	0.07	0.
Causal Oversimplification	0.	0.	0.
Whataboutism	0.	0.08	0.
Loaded Language	0.39	0.62	0.55
Exaggeration/Minimisation	0.11	0.	0.
Repetition	0.40	0.36	0.
Thought-terminating cliché	0.	0.28	0.
Name calling/Labeling	0.39	0.58	0.54
Appeal to authority	0.	0.38	0.22
Black-and-white Fallacy/Dictatorship	0.	0.04	0.
Obfuscation, Intentional vagueness, Confusion (Straw Man)	0.	0.	0.
Reductio ad hitlerum	0.	0.	0.
Appeal to fear/prejudice	0.	0.05	0.
Misrepresentation of Someone's Position (Straw Man)	0.	0.	0.
Flag-waving	0.	0.29	0.
Slogans	0.	0.37	0.04
Doubt	0.25	0.16	0.1

Table 4: Macro F1 scores for different persuasion classes for the given languages for Subtask 1

For Task 1, we started experimenting with the BERT and RoBERTa models, achieving a hierarchical F1 score of 0.55 and 0.60 on the test set of the English language. But, in this approach, we did not take the hierarchy and the definitions of the classes into consideration. We tried to accommodate that using the combination of HypEmo and CDP models.

For the HypEmo model, the model was trained to prioritize higher-level labels in the Directed Acyclic Graph (DAG). During this process, we explored two options: eliminating children when the model predicted the parent label and retaining the children. We observed a significant impact on the hierarchical F1 score, with the first formulation

Language	Base F1	Hierarchical F1	Hierarchical Precision	Hierarchical Recall
English	0.37	0.60	0.53	0.69
Arabic	0.37	0.42	0.32	0.60
Bulgarian	0.28	0.48	0.40	0.62
North-Macedonian	0.30	0.41	0.33	0.56

Table 5: Hierarchical-F1 scores computed across four languages of the test set for sub-task 1. Base F1 score here is the Baseline F1 score

Language	Baseline F1	Hierarchical F1	Hierarchical Precision	Hierarchical Recall
English	0.44	0.67	0.67	0.67
Arabic	0.57	0.53	0.50	0.57
Bulgarian	0.50	0.65	0.66	0.63
North-Macedonian	0.55	0.67	0.72	0.62

Table 6: Hierarchical-F1 scores calculated for four languages within the test set for sub-task 2, with Base-F1 denoting the Baseline F1 score depicted on the leaderboard

yielding 0.45 F1 and the second approach resulting in 0.59 on the test set. We also tried to predict the labels utilizing only the definitions of the classes, using the CDP model, which yielded a hierarchical F1 score of 0.57 and 0.59 on the dev set and the test set, respectively.

For constructing an ensemble, one approach considered concatenating embeddings or softmax predictions from both models for further classification using a neural network. However, this approach was not viable due to limited samples for generalization. The most effective model emerged from utilizing the ensemble with fine-tuning of hyperparameters. Combining predictions from both models yielded a hierarchical F1 score of 0.60.

Table 8 shows that the best generalizability across all tasks is achieved via the HypEmo + CDP(Union) for subtask1.

For sub-task 2, we trained the model from scratch after including the two labels used in the ensemble used in sub-task 1 and changed the feature embeddings being trained by considering the features from the visual content. However, as you can see, there is very little to no difference between the results using CLIP and not using CLIP. We can also see that, unlike the first subtask, they perform better due to more data.

We can see the F1-score analysis tables for each subtask, i.e., in Table 4, Table 3 for subtask1 and subtask2.

For sub-task 3, we trained the model on an ensemble of BERT and CNN models to consider the textual and visual features. It was seen that the ensemble performs just slightly better than using the BERT model, that is, considering only the textual cues. Visual cues are considered significantly when persuasion techniques like $Smears$ are used, as seen in sub-task 2. For the rest of the persuasion techniques, the visual cues were seen not to make a significant impact on the classification task. On the gold labels of the dev set, the ensemble gave a macro-F1 score of 0.67, which is a slight im-

Model	English	Arabic	Bulgarian	North-Macedonian
BERT	0.55	0.39	0.40	0.36
RoBERTa	0.60	0.37	0.45	0.38
HypEmo	0.55	0.43	0.42	0.39
CDP	0.59	0.40	0.48	0.43
HypEmo + CDP (Union)	**0.60**	**0.42**	**0.48**	**0.41**

Table 7: Hierarchical-F1 scores calculated for four languages within the test set for sub-task 1 across different models

Model	English	Arabic	Bulgarian	North-Macedonian
HypEmo (Without CLIP)	0.63	0.511	0.58	0.63
HypEmo (With CLIP)	0.63	0.49	0.59	0.62
CDP	0.64	0.51	0.62	0.65
HypEmo + CDP (Union)	**0.67**	**0.53**	**0.65**	**0.67**

Table 8: Hierarchical-F1 scores calculated for four languages within the test set for sub-task 2 across different models

Language	Base F1	Macro-F1
English	0.25	0.49
Arabic	0.23	0.47
North Macedonian	0.09	0.49
Bulgarian	0.16	0.48

Table 9: Macro-F1 scores computed across 4 languages of the test set for sub-task 3.

Sub-task	Ranking
English-Sub-task1	21
English-Sub-task2	10
English-Sub-task3	19
Bulgarian-Sub-task1	14
Bulgarian-Sub-task2	8
Bulgarian-Sub-task3	11
North Macedonian-Sub-task1	13
North Macedonian-Sub-task2	7
North Macedonian-Sub-task3	7
Arabic-Sub-task1	4
Arabic-Sub-task2	6
Arabic-Sub-task3	13

Table 10: Leaderboard position of our team in the competition in each sub-task

provement from the BERT model, which showed a macro-F1 score of 0.63 on the dev set.

7 Conclusion

Detection of persuasion techniques in memes is seen in a multi-modal setting in this task, but the significant features are drawn from the textual cues in the memes, which can be seen in the results of sub-tasks 1 and 2. The CLIP and other visual language models still need considerable development, and visual cues are helpful for only specific input-output pairs. Identifying whether a persuasion technique is present in the meme but does not apply to the multi-label classification task can be beneficial. Also, we have used a basic ensemble of the latest works in this area and modified them for task-specific requirements. Still, other complex architectures can be explored to get better results.

References

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Chih-Yao Chen, Tun-Min Hung, Yi-Li Hsu, and Lun-Wei Ku. 2023. Label-aware hyperbolic embeddings for fine-grained emotion classification.

Nelson Filipe Costa, Bryce Hamilton, and Leila Kosseim. 2023. Clac at semeval-2023 task 3: Language potluck roberta detects online persuasion techniques in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1613–1618.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Ali Fadel, Ibraheem Tuffaha, and Mahmoud Al-Ayyoub. 2019. Pretrained ensemble learning for fine-grained propaganda detection. In *Proceedings of the second workshop on natural language processing for internet freedom: censorship, disinformation, and propaganda*, pages 139–142.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*, pages 1646–1655. PMLR.

Maria Glenski, Ellyn Ayton, Josh Mendoza, and Svitlana Volkova. 2019. Multilingual multimodal digital deception detection and disinformation spread across social platforms. *arXiv preprint arXiv:1909.05838*.

Bin Guo, Yasan Ding, Lina Yao, Yunji Liang, and Zhiwen Yu. 2020. The future of false information detection on social media: New perspectives and trends. *ACM Computing Surveys (CSUR)*, 53(4):1–36.

Kshitij Gupta, Devansh Gautam, and Radhika Mamidi. 2021. Volta at semeval-2021 task 6: Towards detecting persuasive texts and images using textual and multimodal ensemble.

Vansh Gupta and Raksha Sharma. 2021. NLPIITR at SemEval-2021 task 6: RoBERTa model with data augmentation for persuasion techniques detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1061–1067, Online. Association for Computational Linguistics.

Ivan Habernal, Patrick Pauli, and Iryna Gurevych. 2018. Adapting serious game for fallacious argumentation to german: Pitfalls, insights, and best practices. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Vishal Keswani, Sakshi Singh, Suryansh Agarwal, and Ashutosh Modi. 2020. IITK at SemEval-2020 task 8: Unimodal and bimodal sentiment analysis of Internet memes. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1135–1140, Barcelona (online). International Committee for Computational Linguistics.

Svetlana Kiritchenko, Richard Nock, and Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. volume 4013, pages 395–406.

Patricia Moravec, Randall Minas, and Alan Dennis. 2018. Fake news on social media: People believe what they want to believe when it makes no sense at all. *SSRN Electronic Journal*.

Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Gargi Singh, Dhanajit Brahma, Piyush Rai, and Ashutosh Modi. 2021. Fine-grained emotion prediction by modeling emotion definitions. In *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–8, Los Alamitos, CA, USA. IEEE Computer Society.

Gargi Singh, Dhanajit Brahma, Piyush Rai, and Ashutosh Modi. 2023. Text-based fine-grained emotion prediction. *IEEE Transactions on Affective Computing*, pages 12–12.

Paramansh Singh, Siraj Sandhu, Subham Kumar, and Ashutosh Modi. 2020. newsSweeper at SemEval-2020 task 11: Context-aware rich feature representations for propaganda classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1764–1770, Barcelona (online). International Committee for Computational Linguistics.

Gabriella Szabo. 2020. Emotional communication and participation in politics. *Intersections*, 6:5–21.

Rúben Teimas and José Saias. 2023. Detecting persuasion attempts on social networks: Unearthing the potential of loss functions and text pre-processing in imbalanced data settings. *Electronics*, 12(21):4447.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Shehel Yoosuf and Yin Yang. 2019. Fine-grained propaganda detection with fine-tuned bert. In *Proceedings of the second workshop on natural language processing for internet freedom: censorship, disinformation, and propaganda*, pages 87–91.

HIT-MI&T Lab at SemEval-2024 Task 6: DeBERTa-based Entailment Model is a Reliable Hallucination Detector

Wei Liu[1], Wanyao Shi[2], Zijian Zhang[1], Hui Huang[1]*
[1]Harbin Institute of Technology, Harbin, China
[2]Northwest Normal University, Lanzhou, China
liuweihit2023@163.com, shiwanyao@qq.com,
zhangzj0318@qq.com, huanghui@stu.hit.edu.cn;

Abstract

This paper describes our submission for SemEval-2024 Task 6: SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes. We propose four groups of methods for hallucination detection: 1) Entailment Recognition; 2) Similarity Search; 3) Factuality Verification; 4) Confidence Estimation. The four methods rely on either the semantic relationship between the hypothesis and its source (target) or on the model-aware features during decoding. We participated in both the model-agnostic and model-aware tracks. Our method's effectiveness is validated by our high rankings 3rd in the model-agnostic track and 5th in the model-aware track. We have released our code on GitHub.[1]

1 Introduction

In tasks related to natural language generation, the output of a model may be fluent but may suffer from inaccuracies or inconsistencies with the input, a phenomenon referred to as "hallucination." For instance, Lee et al. (2021) and Müller et al. (2020) noted that in machine translation tasks, translated text is regarded as a "hallucination" when it exhibits a complete disconnect from the source text. Such discrepancies can mislead users and potentially lead to severe consequences. However, current evaluation metrics such as perplexity and BLEU (Papineni et al., 2002a) concentrate more on fluency rather than the accuracy or fidelity to the original input. Therefore, hallucination detection poses a big challenge and has gathered attention from research community.

SemEval-2024 Task 6 (Mickus et al., 2024) presents a testbed to evaluate whether the model outputs are hallucinating or not. The task comprises a total of three kinds of subtasks, which are definition modeling (DM) (Noraset et al., 2017), machine translation (MT) and paraphrase generation (PG). Each subtask involves triplet data with a source, which is the input to the model; a target, which represents the "gold" text that the model is expected to produce; a hypothesis, which is the actual output of the model. For all subtasks, the objective is to evaluate whether the hypothesis exhibits hallucinations according to the source or the target. More specifically, the hallucination of the hypothesis is verified based on target for DM and MT tasks, and source for PG task.

This paper presents the participation of HIT-MI&T Lab in the shared task in detail. We introduce four distinct hallucination detection methods, which transform the problem into different tasks:

1) Entailment Recognition: Hallucination is determined by analyzing the entailment relationship between the hypothesis and its source (target). Our approach mainly involves fine-tuning large language models (LLMs) and DeBERTa (He et al., 2020). An annotation dataset is constructed automatically to address data scarcity. We also devise an optimized loss function to handle noisy annotations during the fine-tuning of DeBERTa.

2) Similarity Search: Hallucination is gauged based on the semantic similarity between the hypothesis and its source (target). We mainly leverage SBERT (Reimers and Gurevych, 2019) to derive sentence representations for similarity search.

3) Factual Verification: Hallucination is detected by identifying factual inconsistencies between the hypothesis and its source (target). We mainly employ UniEval (Zhong et al., 2022) to assess the factual consistency.

4) Confidence estimation: Hallucination is evaluated based on the model's confidence in its answer. We mainly rely on two methods to estimate the model's confidence: a) analyzing the softmax distribution during decoding; b) assessing prediction consistency among multiple samplings.

*Corresponding author.
[1]https://github.com/LiuWeiHITees/semeval2024-task6-hallucination-detection

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1788–1797

Finally, different groups of methods are ensembled for further enhancement, based on the accessibility of model-aware features. With our proposed framework, we achieved the third position in the model-agnostic track and the fifth position in the model-aware track, validating its effectiveness.

2 Related Work

With the success of ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023), natural language generation has gained significant prominence within the broader domain of artificial intelligence. Its applicability spans a diverse array of tasks, including machine translation, summarization, and story continuation, etc. However, these models are sometimes prone to generating outputs that are fluent yet factually inaccurate, a phenomenon referred to as "hallucination". This phenomenon poses a substantial challenge to the reliability of language generation in real-world scenarios.

In the domain of hallucination detection methods, there has been considerable work by predecessors. Some people rely on semantic similarity measures for detection, such as N-gram-based Metrics (ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002b)). However, these metrics only evaluate the lexical overlap between generated texts and reference texts by measuring the n-gram co-occurrence, and cannot discern fine-grained contextual semantic mismatch. Other studies(Laurer et al., 2023; Zha et al., 2023; Vectara, 2023) have fine-tuned BERT models using entailment datasets. These fine-tuned models are then utilized to detect hallucinations in specific scenarios. However, the fine-tuning process requires annotation data and can not generalize well among different scenarios.

As the hallucination mainly comes from the decoding procedure, some people propose to rely on uncertainty measures to detect hallucination. Some research (Guerreiro et al., 2023; Fu et al., 2023) proposed to calculate the log-probability or its entropy of translations for language generation tasks, and a lower probability indicates a lack of confidence, suggesting a potential hallucination. However, access to token-level probability distributions, essential for these approaches, is limited to open-source models and unavailable for models accessed solely through APIs, such as GPT-4.

Recently, with the popularization of LLMs, several LLM-based methods have been proposed. SelfCheckGPT (Manakul et al., 2023), employs a sampling-based strategy, which involves the generation of multiple stochastic samples. This approach hypothesizes that a model with a good understanding of the concept is less likely to generate significant hallucinations. Mündler et al. (2023) has explored the examination of self-contradiction within the context generated by an LLM as another aspect of hallucination detection. Their experiments, which involved prompting the LLM to perform a detection task, have demonstrated successful detection across various LLMs.

3 Methods

In this section, we will introduce our proposed four groups of methods for hallucination detection. The overall framework is shown in Figure 1.

3.1 Entailment recognition

While the objective of hallucination detection is to discern whether there is semantic mismatch between the hypothesis and the source (target), it resembles the objective of entailment recognition. Therefore, we decide to leverage entailment recognition models for detection.

3.1.1 LLM-based Data Construction

When employing entailment recognition model for hallucination detection, task specific fine-tuning is necessary to cope with the domain difference. However, organizers only provide unannotated training data in the form of [source, target, hypothesis], which cannot be directly leveraged for fine-tuning. Therefore, we propose deriving entailment annotations ourselves, leveraging the intelligence of proprietary LLMs like GPT-4. Specifically, we provide the paired text to the LLM, and design the prompt template to utilize GPT-4 to detect hallucinations in the hypothesis[2].

3.1.2 Fine-Tuning DeBERTa

As entailment recognition is inherently a text classification problem, we believe encoder-only understanding models may be more suitable. Therefore, we propose to apply fine-tuning on DeBERTaV3 (He et al., 2021), which has achieved good results especially in the text entailment task. The hypothesis, combined with the source (target) is fed to the entailment model, and a binary label is derived as the detection result.

[2]The detailed prompt is shown in Appendix A due to space limitations.

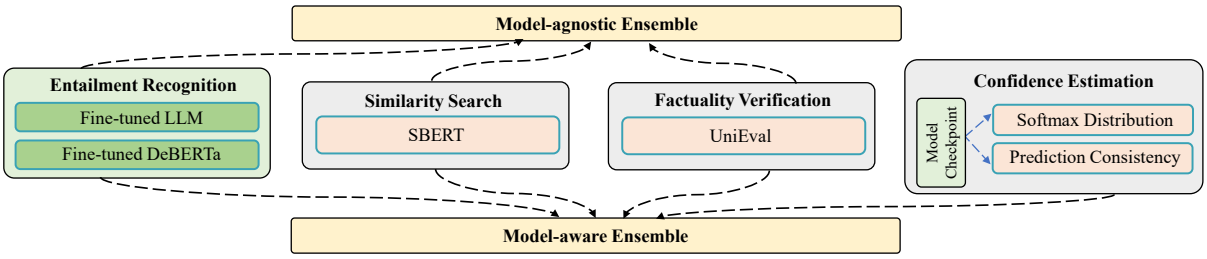


Figure 1: Overall framework of our proposed hallucination detection methods. We ensemble different groups of methods in different tracks, depending on the accessibility of model checkpoints.

As we mainly rely on automatically annotated data for fine-tuning, which the labels are generated by GPT-4 and may contain noise. Therefore, we introduced an auxiliary confidence loss that considers both the annotated labels and the difference between the model's prediction and its own confidence, following the work on weak-to-strong supervision by Burns et al. (2023). The optimized loss is formulated as follows:

$$L_{\mathbf{conf}}(f) = (1-\alpha)\cdot \mathrm{CE}(f(x), f_d(x)) \\ +\alpha \cdot \mathrm{CE}(f(x), \widehat{f}_t(x)) \quad (1)$$

with symbols denoted as follows:

- $\mathrm{CE}(\cdot,\cdot)$ is the cross-entropy loss between the ground truth labels and the predicted probabilities.
- $f(x)$ belongs to [0,1] and represents the model's prediction distribution for input x.
- $f_d(x)$ represents the label for the input x.
- α is a weight used to balance the two losses.
- $\hat{f}_t(x)$ is a special version of $f(x)$, defined as follows:

$$\hat{f}_t(x) = \begin{cases} 1 & \text{if } f(x) > t \\ 0 & \text{if } f(x) \leq t \end{cases} \quad (2)$$

3.1.3 Fine-Tuning LLM

Given the superior performance of open-source LLMs across a diverse array of tasks, we also employ LLM for hallucination detection, which is fine-tuned on the annotated data.

We employ the recently released InternLM-20B (Team, 2023), due to its superior performance across various benchmarks and relatively modest parameter count. Our fine-tuning follows the instruction fine-tuning process, where the hypothesis combined with the source (target) is fed to the model to yield predictions indicative of entailment. Notice as the InternLM has gained massive linguistic knowledge, we only perform fine-tuning on the human annotated validation set. Moreover, we employ Q-LoRA (Dettmers et al., 2023), a parameter-efficient fine-tuning method to reduce the demand for training resources and time.

3.2 Similarity search

Since the hallucination mainly signifies the semantic mismatch between the hypothesis and the source (target), we believe that the mismatch can also be measured by sentence similarity. With contextual sentence embedding models, the hallucination can be discerned by a delicately designed threshold.

Specifically, we use SBERT to derive the semantic representations. SBERT model is an adapted version of BERT (Devlin et al., 2018) which is specifically designed to extract contextual text embeddings. In this work, we construct the sentence representations using the SBERT models for both hypothesis and source (target). After that, the cosine similarity scores are then calculated between the representations, to measure their semantic similarity.

3.3 Factual verification

As hallucinations often relate to factuality contradiction, we think the hallucination can be determined by evaluating the factual consistency between the hypothesis and the source (target).

Specifically, we use the UniEval framework to calculate the factual consistency score. UniEval is a comprehensive framework designed to evaluate generated text across multiple explainable dimensions, including factual consistency assessment. We feed the combined hypothesis and source (target) to the UniEval framework, and a continuous score is derived indicating the factual consistency.

3.4 Confidence Estimation

Hallucination in model output often signifies a lack of confidence. Therefore, we propose to apply

confidence estimation techniques to detect hallucinations. By quantifying the model's confidence in its predictions, we can discern whether the output contains hallucination or not.

Specifically, we employ two confidence estimation techniques: analyzing softmax distribution of output tokens and assessing prediction consistency among multiple samplings. It is important to note that these methods are used only in the model-aware track due to the inherent requirement for checkpoints of the model that generates the output.

3.4.1 Softmax Distribution

One way to estimate model confidence is to analyze the softmax distribution over the vocabulary during the generation process. If the probability mass is highly concentrated on a few words, this suggests the model is confident in its predictions. Conversely, if the softmax probabilities approach a uniform distribution, where picking any word from the vocabulary is equally likely, then the quality of the hypothesis is expected to be low with hallucinations included. Therefore, we propose to incorporate the softmax distribution for hallucination detection.

In particular, we use two groups of features: token-level probability and entropy. For token-level probability, we calculate the average probability and minimum probability of each token. For entropy, we calculate the average entropy and maximum entropy at each position.

This method is primarily applied to DM and MT tasks, as these two subtasks tend to produce fixed outputs for fixed inputs.

3.4.2 Prediction Consistency

When the model lacks confidence with its own prediction, different predictions among different samplings might differ a lot. Based on this premise, we resort to the work of SelfCheckGPT (Manakul et al., 2023), using the model itself to quantify the confidence of the prediction among multiple samplings, thereby detecting hallucinations.

Specifically, we first invoke the model to generate n drawn samples S^n. For the hypothesis and the i-th S^i sample, we invoke the prompt to query LLM and discern their consistency. After that, the hallucination probabilities can be calculated as $\sum_{i=1}^{n} x_i$, with the result x_i for each sample i mapped to a value between 0 and 1. If most of the samples are consistent with the original hypothesis, then the model is confident with its own prediction, and the hypothesis is likely not hallucinated, and vice versa.

We apply this method mainly to the PG task, as this task tends to produce different outputs for fixed inputs. Notice this method does not require the accessibility of glass-box features such as softmax distribution.

4 Experiments

4.1 Experiment-Setup

4.1.1 Data

As shown in Table 1, the organizers provided a validation set with manual annotations and an unannotated training set. As described in Section 3.1.2, for the entailment recognition method, we explore using LLMs like GPT-4 to automatically annotate the unannotated training data. For similarity search, factuality verification, and confidence estimation methods, we mainly rely on the validation set.

Dataset	Track	Task	Quantity
training	model agnostic	DM	1000
		MT	750
		PG	1000
		Total	2750
validation	model agnostic	DM	187
		MT	187
		PG	125
		Total	499
	model aware	DM	188
		MT	188
		PG	125
		Total	501

Table 1: Data Statistics

4.1.2 Pretrained Checkpoints

Regarding the DeBERTa-based entailment model, we mainly rely on DeBERTa-MoritzLaurer, which has already been trained on a diverse range of entailment datasets. For the InternLM-based entailment model, we utilize both the un-instructed and instructed tuned versions for comparison. To derive sentence embeddings from SBERT, we employ three high-performing variants from the text embedding leaderboard[3]. The specific links for all incorporated models are provided in Appendix B.

4.1.3 Task Tracks

This shared task is divided into two tracks: model-agnostic and model-aware. The former operates

[3] https://huggingface.co/spaces/mteb/leaderboard

without knowledge of the hypothesis-generating model. The latter, on the other hand, is informed about the model and can access its checkpoints.

4.2 Main Results

The experimental results are shown in Table 2. The following is a detailed analysis for both model-agnostic and model-aware tracks.

1) DeBERTa-based entailment model performs the best on hallucination detection.

As can be seen, among the four groups of methods, the entailment recognition model performs the best on hallucination detection, across both model-agnostic and model-aware tracks, especially DeBERTa-based entailment model. Although DeBERTa is 50 times smaller than InternLM, it generally outperforms InternLM, possibly due to its encoder-only structure being well-suited for language understanding tasks. Additionally, as the DeBERTa we used is pre-finetuned on various entailment datasets, knowledge can be transferred from other datasets to boost its performance.

Interestingly, the un-instruction tuned InternLM performs better than its instruction tuned version. This indicates the instruction tuning process is inconsistent with our objective and may cause catastrophic forgetting.

2) Similarity-based and factuality-based methods underperform.

In contrast to the entailment-based approaches, similarity-based and factuality-based approaches markedly underperform, potentially due to their mismatches with hallucination detection.

Regarding the similarity-based model, hallucinated sentences might still be similar in the embedding space, as SBERT can only provide general semantic representations. Besides, the Siamese architecture of SBERT also disables in-depth interaction between the source (target) and hypothesis within the multi-layer neural network.

As for the factuality-based model, it mainly aims to evaluate the factual consistency between the source (target) and the hypothesis text, which is a broader task than detecting specific hallucinations. Hallucinations can sometimes be factually consistent with the source information but still contain invented details or distortions, which UniEval's factuality evaluation may not be sensitive enough to capture, leading to poor performance.

3) Confidence estimation performs noticeably worse than other methods.

In the model-aware track, we employ confidence estimation method across all subtasks. We found that this method performed poorly in terms of acc and rho for the DM and MT subtasks. It achieved good acc but poor rho for the PG subtasks. Overall, confidence estimation performed noticeably worse. This can be attributed to two main reasons:

a) The softmax distribution contains insufficient information. The softmax distribution provides a probability distribution over the output vocabulary, but it may not capture all the nuances and uncertainties present in the model's predictions, especially when it comes to hallucinated content.

b) The model fails to provide an accurate evaluation for its prediction. As the prediction is made by the model itself, it is unable to provide an accurate evaluation for the consistency. The consistency verification can only be achieved with the help of external resources among multiple samplings.

Therefore, relying solely on confidence estimation methods may not be effective in detecting hallucinations, as the model itself can be overconfident for its hallucinated outputs.

4) Ensemble of multiple models can enhance performance to some extent.

In the model-agnostic track, the ensembled model achieves an improvement of 0.6 points in acc and 1.5 points in rho. However, in the model-aware track, while the ensemble model surpasses the performance of most models, it is slightly inferior to the best result of the DeBERTa model. We think this might be due to the underperformance of some ensembled methods.

4.3 Analysis of the DeBERTa-based Entailment Model

1) Rationale for not directly utilizing the DeBERTa in entailment model.

As mentioned before, we adopted DeBERTa-MoritzLaurer which is pre-finetuned on various entailment datasets rather than the original DeBERTa model for entailment-based methods. To verify the effectiveness of the pre-finetuning, we perform an ablation study based on the training set and SNLI (Bowman et al., 2015). As can be seen in Table 4, if directly fine-tune DeBERTa on either training set or SNLI, the accuracy on the validation set can achieve only 70%. However, we observe significant performance improvement by employing a two-stage fine-tuning approach, using these datasets sequentially.

Model Type	Model	Description	model-agnostic		model-aware	
			acc	rho	acc	rho
Baseline	Mistral-7B	not train	69.66	40.29	74.53	48.78
Entailment Recognition	InternLM2-20B	train	78.86	67.30	78.20	62.70
	InternLM2-20B-sft	train	63.53	50.35	64.86	46.77
	DeBERTa-MoritzLaurer	train and loss optimization	**82.46**	**75.20**	**80.46**	**71.23**
Similarity Search	SBERT	not train	76.80	63.73	75.66	62.65
Factuality verification	UniEval	not train	72.00	58.04	73.13	54.43
Confidence Estimation	Softmax Distribution	for DM task			59.07	26.08
	Softmax Distribution	for MT task			66.07	37.87
	Prediction Consistency	for PG task			81.33	7.91
Ensemble		ensemble all model	**83.06**	**76.77**	79.73	72.37

Table 2: Experimental results were compared with the baseline of prompting the Mistral-7B model. We use accuracy (which is abbreviated as acc) as the primary evaluation metric and employ Spearman's correlation (which is abbreviated as rho) to comprehensively assess our model's performance.

Model Type	Model	Description	model-agnostic		model-aware	
			acc	rho	acc	rho
Baseline	Mistral-7B	not train	69.66	40.29	74.53	48.78
Entailment Model	DeBERTa-MoritzLaurer	not train	78.00	67.96	63.26	8.21
	DeBERTa-MoritzLaurer	train	81.20	**75.80**	80.13	**71.65**
	DeBERTa-MoritzLaurer	train separately for each task	79.00	66.60	76.40	60.27
	DeBERTa-MoritzLaurer	train and loss optimization	**82.46**	**75.20**	**80.46**	**71.23**

Table 3: Results of different DeBERTa-based entailment models with the following configurations: 1) no training, using the pre-trained model directly; 2) direct fine-tuning using cross-entropy loss; 3) separate fine-tuning on each subtask using cross-entropy loss; 4) fine-tuning with loss optimization.

Model	Description	acc
DeBERTa	fine-tuning on training set	71.23
DeBERTa	fine-tuning on SNLI	72.12
DeBERTa	two stage fine-tuning	78.21

Table 4: DeBERTa model's performance with different fine-tuning settings.

Therefore, instead of directly utilizing the original DeBERTa model, we opted for models pre-finetuned on entailment tasks. Specifically, we selected the DeBERTa-MoritzLaurer model, which is pre-trained on 33 entailment-related datasets, leveraging its transferable entailment recognition knowledge for effective hallucination detection.

2) Loss optimization improves the fine-tuning on LLM-annotated data.

To demonstrate the effectiveness of our proposed loss optimization method, we contrast it with various training methods. As shown in Table 3, while the original model can perform detection to some extent, fine-tuning on annotated data improved the performance. Based on that, our proposed loss optimization method takes into account not only the label but also the model's prediction situation, effectively mitigating overfitting, thereby further improving the performance.

5 Conclusion

In this study, we aimed to address the hallucination detection problem in SemEval-2024 Task 6. We established an ensemble model that includes entailment recognition, similarity search, and factuality verification models. For the model-aware track, we further leveraged confidence estimation for augmentation. Our approach proved effective as we ranked 3rd in the model-agnostic track and 5th in the model-aware track.

Although several methods were incorporated in our experiments, we realized that the best result was achieved primarily by relying on the DeBERTa-based entailment model. Given its portability and generalizability, we plan to further explore its use in hallucination detection in the future.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2023. Building Efficient Universal Classifiers with Natural Language Inference. ArXiv:2312.17543 [cs].

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: learning to define word embeddings in natural language. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3259–3266. AAAI Press.

OpenAI. 2022. Chatgpt blog post. `https://openai.com/blog/chatgpt`.

OpenAI. 2023. Gpt-4 technical report. `https://www.openai.com/gpt4/`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002a. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002b. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.

Vectara. 2023. Hallucination evaluation model.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*.

A The Prompt of GPT-4

Figure 2 shows the specific prompt for asking GPT-4 to perform dataset annotation tasks.

B Utilized Model and Its URL

Table 5 shows the specific model and the corresponding download URL for the utilized model.

Model	URL
InternLM	https://huggingface.co/internlm/internlm2-20b https://huggingface.co/internlm/internlm2-chat-20b-sft
DeBERTa	https://huggingface.co/MoritzLaurer/deberta-v3-large-zeroshot-v1.1-all-33 https://huggingface.co/vectara/hallucination_evaluation_model
SBERT	https://huggingface.co/WhereIsAI/UAE-Large-V1 https://huggingface.co/llmrails/ember-v1 https://huggingface.co/BAAI/bge-large-en-v1.5
UniEval	https://github.com/maszhongming/UniEval
DM Task Checkpoint	https://huggingface.co/ltg/flan-t5-definition-en-base
MT Task Checkpoint	https://huggingface.co/facebook/nllb-200-distilled-600M
PG Task Checkpoint	https://huggingface.co/tuner007/pegasus_paraphrase

Table 5: Details of the utilized model and corresponding download URL

Prompt about task MT:

This is a machine translation task. Given a standard translation, and a model output translation, determine if the model output is subject to hallucination.
your task:
standard translation: {ref}
model output translation: {hyp}
The criteria for judging are as follows:
Check if the model output translation is fluent and answers the question.
Compare the model output translation with correct examples. If inconsistencies are found or it can't be inferred from the standard translation, it's likely hallucination.
If the model output translation aligns with the standard translation or has a similar meaning, it's likely not hallucination.
If the standard translation is "unanswerable" and the model output translation is "I don't know," it's likely not hallucination.
please only return 0 or 1. Return 1 for hallucination; return 0 for not hallucination.

Prompt about task DM:

This is a definition modeling task. Given a standard definition of a word, and a model output definition of this word, determine if the model output is subject to hallucination.
your task:
standard definition: {ref}
model output definition: {hyp}
The criteria for judging are as follows:
Check if the model output definition is fluent and answers the question.
Compare the model output definition with correct examples. If inconsistencies are found or it can't be inferred from the standard definition, it's likely hallucination.
If the model output definition aligns with the standard definition or has a similar meaning, it's likely not hallucination.
If the standard definition is "unanswerable" and the model output definition is "I don't know," it's likely not hallucination.
please only return 0 or 1. Return 1 for hallucination; return 0 for not hallucination.

Prompt about task PG:

This is a paraphrase generation task, which transforms a original sentence into a new sentence. Given a original sentence, and a model output new sentence, determine if the model output is subject to hallucination.
your task:
original sentence: {ref}
model output new sentence: {hyp}
The criteria for judging are as follows:
Check if the model output new sentence is fluent and answers the question.
Compare the model output new sentence with correct examples. If inconsistencies are found or it can't be inferred from the original sentence, it's likely hallucination.
If the model output new sentence aligns with the original sentence or has a similar meaning, it's likely not hallucination.
If the original sentence is "unanswerable" and the model output new sentence is "I don't know," it's likely not hallucination.
please only return 0 or 1. Return 1 for hallucination; return 0 for not hallucination.

Figure 2: The prompt of use GPT-4 to detection hallucination.

UAlberta at SemEval-2024 Task 1: A Potpourri of Methods for Quantifying Multilingual Semantic Textual Relatedness and Similarity

Ning Shi, Senyu Li, Guoqing Luo, Amirreza Mirzaei, Ali Rafiei, Jai Riley, Hadi Sheikhi Mahvash Siavashpour, Mohammad Tavakoli, Bradley Hauer, Grzegorz Kondrak
Alberta Machine Intelligence Institute
Department of Computing Science
University of Alberta, Edmonton, Canada
{ning.shi,senyu,gluo,amirzaei,rafiei,jrbuhr,hsheikhi
siavashp,tavakol5,bmhauer,gkondrak}@ualberta.ca

Abstract

We describe our systems for SemEval-2024 Task 1: Semantic Textual Relatedness. We investigate the correlation between semantic relatedness and semantic similarity. Specifically, we test two hypotheses: (1) similarity is a special case of relatedness, and (2) semantic relatedness is preserved under translation. We experiment with a variety of approaches which are based on explicit semantics, downstream applications, contextual embeddings, large language models (LLMs), as well as ensembles of methods. We find empirical support for our theoretical insights. In addition, our best ensemble system yields highly competitive results in a number of diverse categories. Our code and data are available on GitHub.

1 Introduction

In this paper, we describe our submission for SemEval-2024 Task 1: Semantic Textual Relatedness (STR) (Ousidhoum et al., 2024b), which is based on the SemRel2024 dataset (Ousidhoum et al., 2024a). Each instance consists of a pair of sentences in the same language, annotated with a score that quantifies their semantic relatedness. SemRel2024 was annotated by native speakers of the dataset's 14 languages, which span five language families. An example English instance consists of the sentence pair *"the story is gripping and interesting"* and *"it's a brilliant, compelling, and heartfelt story"*, which is annotated with a relatedness score of 0.64. We participated in all three tracks (supervised, unsupervised, and cross-lingual) on all 14 languages.

Semantic relatedness is distinct from semantic similarity. Sentences that express opposite propositions, such as *"it is raining"* and *"it is not raining"*, exhibit low similarity but high relatedness. The impact of relations such as antonymy and meronymy (Budanitsky and Hirst, 2001) make semantic similarity a more specific task: similarity implies relatedness, but not vice versa. Nevertheless, many traditional algorithms make no attempt to distinguish between the two tasks (Jurafsky and Martin, 2009). For example, the word overlap baseline in this shared task could also be applied to measure semantic similarity. The extent to which semantic similarity and relatedness correlate in practice remains an important open question.

In this paper, we test the hypothesis that *similarity is a special case of relatedness* (Pedersen et al., 2007) through implementing an array of methods that are designed to measure similarity, and applying them to the task of measuring relatedness. We experiment with several different approaches: (1) methods that create and compare semantic representations of each input sentence; (2) methods that use the output of systems designed for other semantic tasks, such as paraphrase identification and entailment detection; (3) methods based on prompting large language models using in-context learning; and (4) methods that combine multiple individual methods. We further posit that *semantic relatedness is preserved under translation*. We investigate both hypotheses via supplementary experiments on datasets from the Semantic Textual Similarity (STS) task at SemEval 2017, as well as new cross-lingual datasets that we construct ourselves by translating parts of the SemRel2024 dataset.

Our experimental results provide support for our theoretical insights. The experiments on the supplementary datasets demonstrate a high correlation between the STR and STS tasks. Out of 51 competing teams, we rank among the top three entries in 16 of the language/track settings. In particular, our best-performing supervised ensemble system achieves the highest score in the English Track A among the teams that submitted a system description paper (Ousidhoum et al., 2024b). Taken together, these results support the idea of using similarity as a proxy for relatedness, as predicted by our hypotheses.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1798–1805

2 Methods

We investigate ten different methods, divided into four types. Each method takes as input a pair of sentences, and produces a scalar value, which, possibly after some normalization to place it within the range specified for this shared task, is used as a measure of STR. Thus, each individual method is a complete, functional STR method; our principal innovation is the ensembling of these methods into a single STR system.

2.1 Explicit Semantic Methods

Concept overlap We hypothesize that the number of shared lexical concepts correlates with the relatedness between two sentences. On the basis of this hypothesis, we tag the words in each sentence with WordNet senses (Miller, 1995) using the offline AMuSE-WSD large model Docker image (Orlando et al., 2021). Each such sense corresponds to a unique lexical concept. The concept overlap score is calculated by dividing the number of shared concepts, with a WordNet synset path similarity greater than 0.8, by the total number of unique concepts in both sentences.

AMR similarity We approximate the relatedness of two sentences by measuring the similarity of their abstract meaning representations (AMRs). The AMR of a sentence is a structured labeled graph that represents its meaning (Banarescu et al., 2013). After converting each input sentence into an AMR using the SapienzaNLP API[1], the similarity between the two AMRs is computed as the Smatch F1-score (Cai and Knight, 2013), a metric devised explicitly for analyzing the overlap between graph-based representations. This score is then used as a measure of the relatedness of the sentences.

2.2 Extrinsic Methods

Paraphrase identification (PI) We reduce STR to paraphrase identification, a binary classification task to determine the approximate semantic alignment between two sentences (Bhagat and Hovy, 2013). By utilizing a dedicated PI model, we first compute the probability that one sentence is a paraphrase of the other. The intuition is that a higher probability of a positive classification indicates greater semantic relatedness. While paraphrasing is, in theory, a symmetric relation on sentences, in practice, the order in which sentences are provided to the model impacts its output. We compute the paraphrase identification probability for both orderings of the two sentences, and use their average as the score for STR.

Taking RoBERTa (Liu et al., 2019) as the backbone, we fine-tune a paraphrase classifier on a combined dataset, including six datasets: PIT (Xu et al., 2015), QQP (Iyer et al., 2017), MRPC (Dolan and Brockett, 2005), PAWS QQP (Zhang et al., 2019), PAWS Wiki (Zhang et al., 2019), and PARADE (He et al., 2020). We follow dataset splits and training configurations as established in prior research (He et al., 2020; Peng et al., 2022).

Textual entailment Similar to PI, we use textual entailment as an indicator of sentence relatedness. In particular, we aim to reduce STR to recognizing textual entailment (RTE) or natural language inference (NLI). Both tasks evaluate whether the meaning of one sentence (the hypothesis) can be inferred from another (the premise). RTE frames this as a binary task and NLI expands it into ternary classification with the addition of a neutral label (Dagan and Glickman, 2004). Recognizing that entailment in either direction signifies potential relatedness, we use an off-the-shelf RoBERTa NLI classifier (Nie et al., 2020) to estimate the probability of entailment in both directions. The final STR score is the average of these two probabilities.

2.3 Distributional Methods

Embeddings In this method, we use an LLM to produce dense semantic embeddings representing the meaning of each input sentence. We then compute the cosine similarity between their respective embeddings, and use this as a measure of relatedness. This simple embedding-based approach allows a language model to be used "as-is", with no need for additional fine-tuning.

We experiment with two variants based on BERT (`Embed-B`) (Devlin et al., 2019) and RoBERTa (`Embed-R`), respectively. For each sentence, hidden states are obtained from the LLM, and an attention mask is applied to ensure the model focuses on meaningful tokens and excludes the other special ones such as the padding token. The resultant hidden states are aggregated into a single vector through average pooling.

2.4 Large Language Models

Prompting We utilize a few-shot prompting strategy to estimate STR between sentence pairs. We

[1] `nlp.uniroma1.it/spring/api/text-to-amr`

use in-context learning (Brown et al., 2020), providing first a small set of examples from the training data, consisting of two sentences and an STR value (i.e., the correct output from the data). For each pair of sentences, To facilitate few-shot prompting, we sample example sentence pairs from the training dataset and query ChatGPT through its API.[2]

Fusion This approach makes use of contextualized embeddings from a variety of open-source LLMs. For each sentence, we extract its sentence embeddings from each LLM, and concatenate them. The result is a "fusion" vector embedding of sentences whose dimensionality is the sum of the dimensionality of the embeddings produced by each LLM. We apply a trainable point-wise linear operation with bias to the fusion embeddings. We train this layer to minimize the distance between the cosine similarity of the fusion embeddings of each sentence pair in the training data and their gold-standard STR scores. In other words, we train this layer to produce the cosine similarity as the STR scores given pairs of fusion embeddings.

We integrate embeddings derived from a range of sentence transformer models (Reimers and Gurevych, 2019). While several of them are multilingual, our training process is exclusively focused on the English dataset. It aims to minimize the mean squared error (MSE) loss between the cosine similarity of the fusion embeddings for sentence pairs and their corresponding gold-standard STR scores. We adopt early stopping to mitigate the risk of overfitting.

Fine-tuning We add a linear regression head to a pre-trained language model, and fine-tune it for STR using the training data. The resulting regression model is therefore optimized for predicting the relatedness score given a pair of sentences. This provides another approach for leveraging the semantic capabilities of modern language models.

We investigate three distinct regression models, with one variant. Each regression model takes an LLM as the backbone with a randomly initialized regression head. We proceed to fine-tune the entire model, both the backbone and the regression head, The AdamW optimizer (Loshchilov and Hutter, 2019) is configured with an initial learning rate of 2e-5 and a batch size of 24. For the backbone, we experiment with T5 (`FT-T5`) (Raffel et al., 2020), GPT-2 (`FT-GPT2`) (Brown et al., 2020), and RoBERTa (`FT-R`). The variant `FT-MPNet` uses MPNet (Song et al., 2020), aligning more with the training process of SBERT. While most models are trained to minimize the MSE loss, MPNet uniquely targets minimizing the cosine similarity loss. This positions the MPNet one as a form of continued pre-training. We categorize this as a variant within our fine-tuning method for better presentation.

[2]GPT-3.5-turbo-1106. Our experiments with LLaMA-2 (Touvron et al., 2023) were unsuccessful.

2.5 Ensemble Modules

To combine the advantages of the methods above, we assess two ensembling strategies: *unsupervised linear combination* and *supervised regression*.

Linear combination Our first approach is to compute the average of the STR scores produced by the individual methods. We first normalize the scores, based on the observation that some methods tend to produce higher or lower scores (i.e., scores with very different distributions). For instance, one method might typically produce scores between 0.7 and 0.9, while another might tend to produce scores in the range of 0.2 to 0.6. Our normalization is intended to give each method a similar distribution, with the lowest scores being normalized to 0 and the highest scores being normalized to 1. Once normalization is complete, for a given pair of sentences, the final ensemble STR score is obtained by computing the average score across all methods.

Our official submission for Track B is a linear ensemble system `Linear-2Ms` applied to synthesize the results of `Embed-B` and `Embed-R`. It operates entirely unsupervised, meaning it does not require exposure to any samples from the training set.

Regression One limitation of the linear combination is that it makes no distinction between methods; each method contributes equally to the average, regardless of how reliable it is in practice. Our second method combines the scores from the individual methods by treating each score as a feature in a linear regressor. Once trained, the method is applied by first computing the outputs of each method in the ensemble, and then applying the regression model to obtain the final score.

Our official submission for non-English languages in Track C, as well as English in Track A, is a regression ensemble system `XGB-4Ms` designed to synthesize the outputs from fine-tuning T5, GPT2, RoBERTa, and MPNet. At the heart of this ensemble system, we deploy an XGBoost regressor (Chen and Guestrin, 2016) as the central

Method	afr	amh	arb	arq	ary	eng	esp	hau	hin	ind	kin	mar	tel	Avg.
Overlap	71.0	63.0	32.0	40.0	63.0	67.0	67.0	31.0	53.0	55.0	33.0	62.0	70.0	54.4
LaBSE-Cross	79.0	84.0	61.0	46.0	40.0	80.0	62.0	62.0	76.0	47.0	57.0	84.0	82.0	66.2
LaBSE-Sup	-	85.0	-	60.0	77.0	83.0	70.0	69.0	-	-	72.0	88.0	82.0	-
WordOvlap	73.2	64.3	31.4	40.2	57.7	73.9	63.5	38.7	57.1	53.2	31.5	68.9	64.3	55.2
EngWordOvlap	74.5	65.9	34.8	42.4	40.9	73.9	61.3	39.6	59.3	49.6	27.5	69.3	71.7	54.7
ConceptOvlap	69.7	60.4	42.6	38.2	40.7	68.8	61.1	34.9	59.0	38.5	30.7	64.7	70.3	52.3
AMR	70.1	62.7	30.7	35.9	33.7	71.4	60.6	36.0	59.5	45.7	33.2	67.8	66.4	51.8
PI	48.2	66.7	32.6	30.6	32.0	73.6	42.6	49.4	63.3	32.4	46.1	71.0	74.3	51.0
NLI	25.9	33.2	21.5	1.5	8.5	64.5	24.4	39.6	57.5	18.3	41.6	64.4	67.3	36.0
Embed-B	77.9	71.3	44.1	36.5	6.0	77.4	67.7	41.9	69.1	46.1	36.4	77.9	71.5	55.7
Embed-R	79.3	71.5	45.9	35.0	11.4	75.2	67.7	32.0	67.0	49.3	38.9	75.5	65.7	55.0
Prompt	80.4	78.2	64.0	40.2	38.1	82.0	62.2	57.6	80.3	47.1	53.0	85.5	82.9	65.5
Fusion	82.5	80.6	70.2	42.9	30.8	84.6	64.1	64.7	80.0	48.8	56.0	84.7	84.1	67.2
FT-T5	78.8	80.5	58.9	35.0	56.1	82.3	53.8	63.3	78.8	39.9	62.2	82.0	84.8	65.9
FT-GPT2	79.4	78.4	54.8	44.4	51.1	82.9	58.7	60.4	75.5	46.0	57.8	80.7	82.3	65.6
FT-R	81.1	80.8	65.7	43.2	54.4	83.6	55.8	67.7	82.3	39.4	64.1	86.6	84.1	68.4
FT-MPNet	81.7	80.3	67.9	44.7	25.3	84.9	64.0	61.4	80.2	52.2	53.5	84.7	82.4	66.4
Linear-2Ms	78.9	72.3	46.7	36.8	8.1	77.5	68.0	38.0	69.1	48.4	37.8	78.0	69.3	56.1
XGB-3Ms	80.6	81.5	67.3	45.1	60.4	84.6	57.0	67.4	82.2	45.6	63.7	85.5	85.2	69.7
XGB-4Ms	81.8	82.1	70.2	47.0	48.0	85.6	60.4	67.4	82.8	49.4	62.1	86.5	86.1	70.0
Target-XGB	-	85.4	-	57.5	80.6	85.6	70.5	73.5	-	-	77.4	89.0	85.7	-

Table 1: The results on the test sets of SemRel2024 in terms of the Spearman correlation (%).

model, tasked with integrating the results of the four individual systems as input features to predict the STR score. The XGBoost regressor is configured with a squared error regression objective and tailored configurations to optimize performance: a column sample by tree of 0.1, a learning rate of 0.1, a maximum depth of 8, an alpha value of 0.1, and 128 estimators. Training will stop if there is no improvement during validation for 32 consecutive rounds. This set of hyper-parameters is optimized on the English development set and remains constant throughout experiments.

Our official submission for non-English languages in Track A is Target-XGB, a tailored variant of the XGB-4Ms system, which is specifically engineered to navigate the linguistic distribution shifts inherent across languages. Underpinning this approach is the assumption that STR between sentence pairs remains consistent across languages. To this end, we fine-tune each individual system and the XGB regressor on English translations within the target language. Recognizing the potential introduction of noise from imperfect machine translation systems, we implement a data augmentation technique. To be specific, we merge the English training and development sets with the translated training set of the target language. The translated development set of the target language is kept as it is out of the training. By that, we intend to treat the English dataset as a stabilizing anchor, mitigating translation noise and ensuring that our system still remains sensitive to the target language.

3 Semantic Textual Relatedness

Our principal evaluation is on Tracks A, B, and C of the shared task datasets. The evaluation results are reported in Table 1, excluding Punjabi (pan), where most results are negative without any observable pattern. Our results may differ from those submitted due to adjustments in methods. We report the Spearman correlation (%) between the prediction and the golden STR scores.

We employ several baseline methods. Overlap, LaBSE-Cross and LaBSE-Sup are the official baselines reported by Ousidhoum et al. (2024a); the key distinction between the last two lies in whether the backbone LaBSE (Feng et al., 2022) is fine-tuned or not. WordOvlap is our re-implementation of Overlap; EngWordOvlap is its variant which requires translating sentences into English first. Most of our systems are English-specific; we translate sentences in other languages into English via the Google Translate API.

Explicit methods ConceptOvlap performs comparably to WordOvlap, indicating a similar level of efficacy in capturing semantic relatedness. However, AMR lags behind, suggesting that representing sentences as semantic graphs may introduce information that does not contribute to determining semantic relatedness, or that the quality of the AMR representations is insufficient.

Extrinsic methods Reducing the STR task to either PI or NLI yields markedly distinct outcomes. While PI approaches the performance level of the WordOvlap baseline, NLI generally underperforms. This discrepancy may be attributed to the inherent unidirectional nature of entailment. Our implementation takes the average of two entailment probabilities, and thus imposes strict constraints on the relatedness of sentences.

Distributional methods We can see that both Embed-B and Embed-R secures commendable results, matching the overall performance of the WordOvlap baseline. The observed superiority of BERT over RoBERTa could stem from differences in their score distributions. The predictions of Embed-R tend to be more clustered (e.g., ranging from 0.84 to 0.99 on eng). We found that simply rounding its results to two decimal places could reduce its performance from 75.2 to 72.4 on eng. In contrast, the predicted score distribution of Embed-B is relatively more dispersed.

LLM methods Prompt is competitive with LaBSE-Cross, but well below other LLM methods, such as Fusion and FT. This shows that, despite its strong performance on many other tasks, ChatGPT's STR capabilities are still limited. Furthermore, training on the provided dataset is observed to be pivotal in enhancing performance. Overall, our findings underline that there remains considerable scope for exploring and enhancing the application of LLMs in this field.

Ensemble modules Target-XGB obtains the best results across most languages. It surpasses LaBSE-Sup and consistently exceeds XGB-4Ms across all evaluated languages by a significant margin. These results show the importance of additional fine-tuning using the translations of the target language. Furthermore, incorporating the English dataset alongside the translated dataset proves to be advantageous. Notably, our ensemble systems, using either linear or regression modules, demonstrate superior performance over the individual systems they comprise, supporting the efficacy of our proposed ensemble approach.

4 Cross-Lingual Textual Relatedness

In this section, we discuss our experiments on new cross-lingual datasets which we created from the shared task data. The purpose of these experiments is to test our hypothesis that *semantic relatedness is preserved under translation*. Our bilingual dataset contains pairs of sentences from English and Spanish, respectively. The Spanish sentences are obtained by alternately translating one of the two English sentences from each instance of the SemRel2024 development set. The task is to determine the cross-lingual STR score, which is assumed to be the same as that for the original monolingual English sentence pair.

Method	eng	esp*	eng-esp	eng-esp*
WordOvlap	62.7	57.8	33.1	62.5
ConceptOvlap	63.4	62.1	51.5	64.1
AMR	66.1	61.4	-	64.2
PI	71.6	40.3	-	71.5
NLI	62.0	38.2	-	61.6
Embed-B	72.4	71.0	-	70.9
Embed-R	72.1	**72.0**	-	67.0
Prompt	79.0	67.5	77.1	78.7
Fusion	82.5	68.2	**80.4**	81.7
XGB-4Ms	**85.6**	68.4	-	**84.9**

Table 2: Results of primary methods evaluated using Spearman correlation (%) in our cross-lingual setting. Translating inputs into English is denoted by "*".

Table 2 shows the experimental results on our cross-lingual STR dataset. The eng-esp column shows the results of those methods that can be directly applied to languages other than English. The eng-esp* column shows the results of a larger subset of methods obtained after translating the Spanish sentence in each instance back into English. For reference, we also include the results on the official English (eng) and Spanish (esp*) development sets, of which the latter is translated into English.

The WordOvlap baseline performs poorly when applied to the eng-esp dataset because orthographic forms rarely match across languages even if they have the same meaning. In contrast, ConceptOvlap performs much better, as it is entirely multi-lingual and independent of orthography and script. However, both methods obtain similar results on eng-esp*, where the Spanish text is translated into English. Our AMR, NLI, and XGB-4Ms systems cannot be applied to cross-lingual pairs, but when Spanish is translated into English, their performance on eng-esp* is comparable to what is observed on the English test set.

The most interesting findings emerge from the results of Prompt and Fusion. Both are applicable directly to cross-lingual data, without translating the Spanish sentences into English. Surprisingly, for both methods, we observe only small differ-

Method	eng-eng	eng-esp	eng-esp*
ECNU	85.2	81.3	-
WordOvlap	72.8	13.5	64.4
ConceptOvlap	74.8	50.3	69.0
AMR	71.5	-	59.2
PI	76.9	-	72.2
NLI	68.4	-	69.7
Embed-B	73.5	-	63.4
Embed-R	71.5	-	59.2
Prompt	89.2	**87.9**	**88.4**
Fusion	90.3	84.4	87.8
XGB-4Ms	**91.0**	-	87.6

Table 3: Evaluation results of our primary methods using Spearman correlation (%) for the STS task. Translating inputs into English is denoted by "*". ECNU (Tian et al., 2017) ranked first in the SemEval 2017 Task 1.

ences between the relatively high numbers in the three columns. This finding supports our hypothesis that translation does not affect the degree of relatedness between a pair of sentences.

5 Semantic Textual Similarity

Another hypothesis that we investigate is that *similarity is a special case of relatedness*. Therefore, we expect a strong correlation between the two concepts: sentences that are highly similar should also be considered highly related. In this section, we test this hypothesis by applying our methods to STS datasets (i.e., track 5 and 4a) from SemEval 2017 Task 1 (Cer et al., 2017).

Since both STR and STS tasks output numerical scores on pairs of sentences, our methods can be directly applied to STS without modification. For supervised methods, we apply the models to STS in the same way as to STR, without any additional training or fine-tuning. This approach can be viewed as transfer learning: models trained on STR datasets are tested on the STS task. For cross-lingual datasets, we again experiment with both direct application to different languages (eng-esp) and pre-translation into English (eng-esp*).

Table 3 shows the results of our STS experiments. While these STS results are not directly comparable to any STR results, we observe that the best three methods are the same for both mono-lingual and bilingual STS and STR. As shown in Figure 1, the progressive improvement from left to right across methods indicates a strong correlation between STR and STS. Furthermore, the overall alignment between the blue and green lines, as well as between the red and yellow lines, support our hypothesis that both STR and STS are preserved

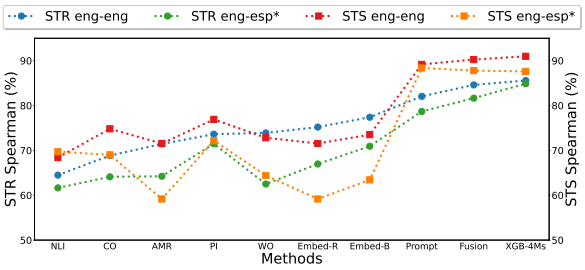


Figure 1: Summary of evaluation results for our primary methods in both STR and STS, Methods are ordered by their performance on STR eng-eng.

under translation.

A more detailed analysis of the individual methods reveals several additional insights. Among the explicit methods, WordOvlap and AMR are both outperformed by ConceptOvlap on STS, which is likely due to their lack of robustness in cross-lingual settings. Among the extrinsic methods, PI works better than NLI, exhibiting remarkable stability across tasks, likely due to the training of our PI model on diverse benchmark datasets, including adversarial examples from PAWS QQP and PAWS Wiki. Among the distributional methods, while Embed-B consistently outperforms Embed-R, both experience a decline in cross-lingual performance, revealing sensitivity of sentence embeddings to translation noise. Among LLMs-based methods, Prompt excels on cross-lingual STS benchmarks, possibly because of data leakage in training ChatGPT, as well as its multilingual design. Our ensemble system XGB-4Ms generally delivers the best results.

6 Conclusion

We have investigated a wide array of methods on two of sentence-level semantic tasks in both mono-lingual and cross-lingual settings. In the process, we assembled a comprehensive benchmark of datasets for future explorations in this domain. The experiments furnish evidence for two hypotheses: (1) semantic similarity is a special case of semantic relatedness, and (2) both similarity and relatedness are preserved under translation. In practical terms, the evaluation results indicate that ensembling LLMs with diverse architectural designs yields the most robust and effective performance across languages and tasks. Notably, our strongest system is at the top of the ranked teams in English Track A, the setting with the highest number of participants.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Alberta Machine Intelligence Institute (Amii).

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Rahul Bhagat and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Alexander Budanitsky and Graeme Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and other lexical resources*, volume 2, pages 2–2.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Ido Dagan and Oren Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. *Learning Methods for Text Understanding and Mining*, 2004(26-29):2–5.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, and James Caverlee. 2020. PARADE: A New Dataset for Paraphrase Identification Requiring Computer Science Domain Knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7572–7582, Online. Association for Computational Linguistics.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. Quora question pairs. *First Quora Dataset Release: Question Pairs*.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing*, 2nd edition. Prentice Hall.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

George A Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Riccardo Orlando, Simone Conia, Fabrizio Brignone, Francesco Cecconi, and Roberto Navigli. 2021. AMuSE-WSD: An all-in-one multilingual system for easy Word Sense Disambiguation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 298–307, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. Semeval task 1: Semantic textual relatedness for african and asian languages.

Ted Pedersen, Serguei V.S. Pakhomov, Siddharth Patwardhan, and Christopher G. Chute. 2007. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, 40(3):288–299.

Qiwei Peng, David Weir, Julie Weeds, and Yekun Chai. 2022. Predicate-argument based bi-encoder for paraphrase identification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5579–5589, Dublin, Ireland. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 task 1: Leverage kernel-based traditional NLP features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 191–197, Vancouver, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

HW-TSC at SemEval-2024 Task 5: Self-Eval? A Confident LLM System for Auto Prediction and Evaluation for the Legal Argument Reasoning Task

Xiaofeng Zhao[1], **Xiaosong Qiao**[1], **Min Zhang, Chang Su, Yuang Li, Yinglu Li, Yilun Liu Feiyu Yao, Xiaowei Liang, Shimin Tao, Hao Yang, Yanfei Jiang, Yunfei Lu, Dandan Tu**
Huawei Translation Services Center, Beijing, China
{zhaoxiaofeng14, qiaoxiaosong, zhangmin186, suchang8, liyuang3, liyinglu, liuyilun3, yaofeiyu1, liangxiaowei2, taoshimin, yanghao30, jiangyanfei, luyunfei6, tudandan}@huawei.com

Abstract

In this article, we present an effective system for semeval-2024 task 5. The task involves assessing the feasibility of a given solution in civil litigation cases based on relevant legal provisions, issues, solutions, and analysis. This task demands a high level of proficiency in U.S. law and natural language reasoning. In this task, we designed a self-eval LLM system that simultaneously performs reasoning and self-assessment tasks. We created a confidence interval and a prompt instructing the LLM to output the answer to a question along with its confidence level. We designed a series of experiments to prove the effectiveness of the self-eval mechanism. In order to avoid the randomness of the results, the final result is obtained by voting on three results generated by the GPT-4. Our submission was conducted under zero-resource setting, and we achieved first place in the task with an F1-score of 0.8231 and an accuracy of 0.8673.

1 Introduction

In 2023, a significant event in the field of artificial intelligence (AI) was the widespread adoption of ChatGPT, particularly the introduction of GPT-4 (OpenAI, 2023), which revolutionized perceptions of AI. GPT-4 exhibited a notable advancement of 11.2 points on the MMLU benchmark (Hendrycks et al., 2021) and demonstrated superior performance on various question answering (QA) and natural language inference (NLI) datasets. Large-scale language models (LLM) represented by GPT-4 have sprung up, including LLaMa-2 (Touvron et al., 2023), Falcon (Almazrouei et al., 2023), Gemini (Anil et al., 2023), Baichuan-2 (Yang et al., 2023) , ChatGLM (Du et al., 2022), etc. There are many researchers have explored NLP task leveraging GPT-4 in zero-resource and low-resource scenarios. GPT-4 is pretrained on a large amount of Internet data initially, and refined through supervised fine-tuning and reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022). Despite these advancements, limited research exists on the direct application of GPT-4 to NLP tasks within the legal domain. This paper aims to comprehensively address this research gap.

2 Task Description

This task aims to assess system's ability to reason about legal arguments. The task organizers have introduced a dataset (Bongard et al., 2022) of civil litigation cases from the U.S. legal system. Each instance comprises of an overview of the case, a question, a proposed solution (an answer candidate), and analysis justifying the solution. Systems are required to determine if the solutions and analysis are correct (True) or incorrect (False). While similar to a typical classification task, this task demands strong causal reasoning and practical application of knowledge. As shown in Figure 1, evaluating the correctness of an answer candidate demands not only a logical assessment of the question and response but also the application of legal knowledge provided in the introduction.

Further, this task requires expertise with legal terminology and concepts. An experienced law professor, armed with deep understanding of relevant legal statutes and extensive knowledge in the field, would likely be able to swiftly assess the accuracy of answer candidates and the soundness of their analysis, even with minimal background information. Conversely, for those less familiar with the field, even being provided with comprehensive information, identifying key details and reaching the correct conclusion remains a challenging task. Notably, the training dataset and development dataset provide analysis of the labels, whereas the test dataset does not.

This dataset is extracted from real law teaching books and includes a total of 666 training sets, 84 development sets, and 98 test sets. The training set and development set provide analysis of labels, but

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1806–1810

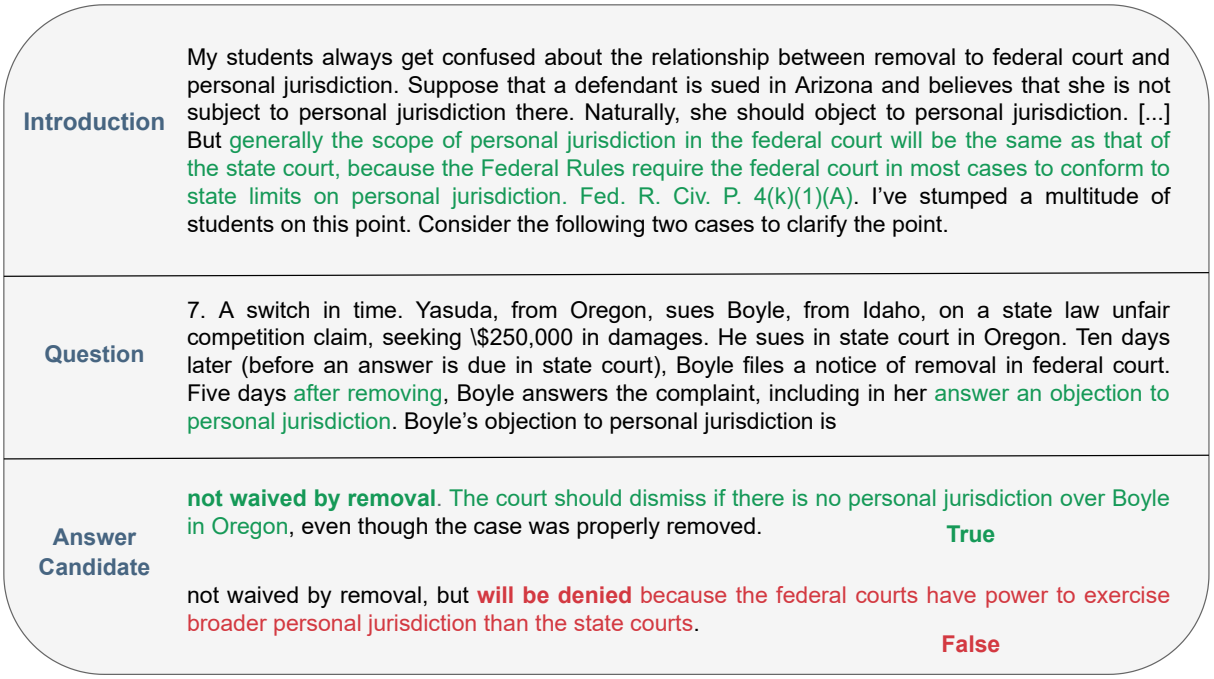


Figure 1: Data Example

the test set is not provided. The goal is to predict the label of the test set.

3 System

3.1 Method Overview

For this task, we have designed a Self-Eval LLM system that utilizes LLM (e.g. GPT-4) for reasoning to obtain answers. However, the responses generated by LLM can sometimes be ambiguous. Therefore, we have incorporated a confidence detection task to enable the LLM to evaluate the reliability of its own answers, and stimulate the LLM's potential. We use one specifically designed prompt for the model to perform both tasks — judging answer candidate and providing answer confidence. Furthermore, we have employed two strategies: converting judgments into selections and ensemble learning. As shown in Figure 2, we depict a workflow with and without confidence.

3.2 Inference with Confidence

This task demands strong causal reasoning skills and specialized knowledge in the legal domain, intutively beyond the capabilities of small models like BERT. LLMs are trained with vast Internet-based corpora, obtaining extensive knowledge and causal reasoning capabilities. Hence, we opted to utilize LLM, specifically the GPT-4, the model name is "gpt-4-0125-preview" and all hyperparameters use the default. Furthermore, in response to the ambiguity in LLM's responses, we proposed the Self-Eval mechanism, wherein the LLM is required to assess the confidence of its answer candidates while generating outputs. Our prompt took the following form: [*You are an AI assistant with reasoning and distinguishing abilities in the legal field. I have a new NLP task and dataset from the domain of the U.S. civil procedure. As a legal assistant, you can help me decide whether the relevant answer is correct or not. I will provide an explanation, an analysis, a question, and an answer. Please analyze to see if the answer is correct and give your confidence on a scale of 0-5, where the higher the score, the more accurate you think your answer is. The output format is: Analysis:, Is correct (Yes/No):, Confidence score:*].

Additionally, we found that LLM performs better in choice tasks than in judgment tasks. By examining task data examples, we noticed that some examples had the same introduction and question. Therefore, we converted data from true or false questions to multiple-choice questions. Specifically, we assigned numbers to answer candidates for LLM to choose from. If all options are incorrect, it returns None. For choice tasks, the prompt format we used was as follows: [*You are an AI assistant with reasoning and distinguishing abilities in the legal field. I have a new NLP task and dataset from the domain of the U.S. civil procedure. As a*

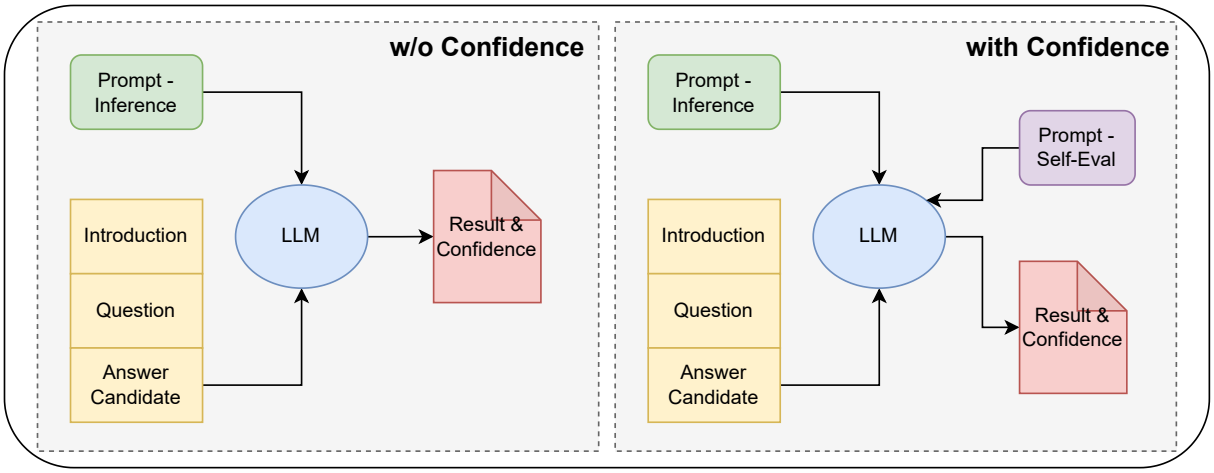


Figure 2: Flowchart of LLM System w/o Confidence and with Confidence

legal assistant, you can help me decide whether the relevant answer is correct or not. I will provide an explanation, an analysis, a question, and a few answers. Only one or none of these answers is correct. Please determine which answer is correct, Note that there may be cases where none of the answers are correct. Give a confidence score (0-5) on the larger model's answer, with higher scores indicating that you think the answer is correct. The output format is: correct answer: answer-id, confidence: score:]. Ultimately, to alleviate the model's stochastic nature, we implemented an ensemble strategy where, for each LLM, we ran it three times and aggregated the inference results, which is the final version we used in the evaluation. Due to cost constraints, we only implemented an ensemble strategy in the experimental group with the highest performance results.

3.3 Inference without Confidence

In addition to the previously mentioned LLM system equipped with the Self-Eval mechanism, we also conducted experiments involving direct inferring. The experimental parameters were kept consistent with the previous settings. When assessing judgement tasks, we utilized the following prompt format: [*You are an AI assistant with reasoning and distinguishing abilities in the legal field. I have a new NLP task and dataset from the domain of the U.S. civil procedure. As a legal assistant, you can help me decide whether the relevant answer is correct or not. I will provide an explanation, an analysis, a question, and an answer. Please analyze to see if the answer is correct. The output format is: Analysis:, Is correct (Yes/No):*]

When tackling choice tasks, the prompt we utilize is as follows: [*You are an AI assistant with reasoning and distinguishing abilities in the legal field. I have a new NLP task and dataset from the domain of the U.S. civil procedure. As a legal assistant, you can help me decide whether the relevant answer is correct or not. I will provide an explanation, an analysis, a question, and a few answers. Only one or none of these answers is correct. Please determine which answer is correct, Note that there may be cases where none of the answers are correct:*]

3.4 2Pass Strategy

In addition to the above experiments, we also designed a 2pass LLM reasoning and evaluation experiment to verify the self-evaluation ability of LLM. We take true or false questions as an example. First, we prompt the LLM to provide reasoning-only answers. Next, we ask the LLM to provide confidence scores for its answers, and gain the final result based on the confidence score. If confidence exceeds 3, we maintain the original result given by the LLM, otherwise we flip it. The prompts for the first pass are the same as the judgment only, and the prompts for the second pass are as follows: [*You are an AI assistant with reasoning and distinguishing abilities in the legal field. I have a new NLP task and dataset from the domain of the U.S. civil procedure. I will provide an explanation, an analysis, a question, and an answer. And analysis and judge of LLM, Give a confidence score (0-5) on the larger model's answer, with higher scores indicating that you think the answer is correct. The output format is: confidence: score:*].

Model	F1 Score	Accuracy
GPT-4-judgement only	0.7061	0.7551
GPT-4-judgement with confidence	0.7211	0.7653
GPT-4-2pass	0.6984	0.7341
GPT-4-choice only	0.7644	0.8163
GPT-4-choice with confidence	**0.8012**	**0.8649**
irene.benedetto's System	0.7747	0.8265
GPT-4-choice with confidence (Ensemble)	**0.8231**	**0.8673**

Table 1: Results of different models for test

4 Results and Analysis

4.1 Overview

Table 1 shows the results of different strategies on the test set of this task, where the representatives not marked with the Ensemble flag only run a single experiment. The evaluation metrics are F1 score and accuracy. As it is shown in the table, our final system, GPT-4-choice with confidence (Ensemble), has achieved the highest scores on both metrics, outperforming the best system from other participants, irene.benedetto, by absolute margins of 4.08 percentage points on F1 score and 2.65 percentage points on accuracy. Even without ensembling, our approach still improves F1 score by 2.65 percentage points and accuracy by 3.84 percentage points. This can prove the effectiveness of our system on the Legal Argument Reasoning task.

Table 1 presents a comparison between the performance of GPT-4 with and without Self-Eval. The results indicate a notable improvement when real-time confidence assessment is implemented. Specifically, for true or false questions, the F1 score of GPT-4 with confidence assessment is 1.50 percentage points higher than that without confidence assessment, with a corresponding 1.02 percentage point increase in accuracy. Similarly, in the case of multiple-choice questions, the F1 score for GPT-4 with confidence assessment outperforms the model without by 3.68 percentage points, while accuracy improves by 4.86 percentage points. This significant enhancement in performance underscores the value of integrating real-time confidence assessment into the model. The observed effect is attributed to the Self-Eval mechanism, which appears to stimulate the latent capabilities of the LLM. By prompting the LLM to evaluate its own confidence levels, it performs reasoning tasks more diligently, resulting in more accurate answers.

From Table 1, we can also see that on F1 score, 2Pass prompting underperforms the judgment-only and judgment-with-confidence versions of GPT-4 by 0.77 and 2.27 percentage points respectively. This suggests that ex-post confidence assessment does not benefit the model's answers. Confidence assessment only improve performance if conducted concurrently with inference. While surprising, this indicates promise for further exploration.

4.2 Case Study

As we mentioned in Section 3.1, the output of LLM is sometimes ambiguous. To address this, we introduced a self-evaluation mechanism. The following case illustrates its impact. As shown in figure 3, before the Self-Eval mechanism was introduced, when posed with a true of false question, the model replied "not entirely correct" — an ambiguous response falling between true and false. In contrast, analysis of multiple responses after incorporating the self-assessment mechanism revealed no ambiguous statements. These results demonstrate the effectiveness of the self-evaluation task at reducing ambiguity in LLM's responses.

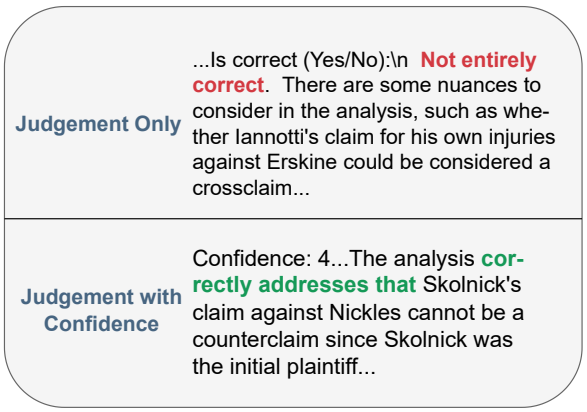


Figure 3: Case about Ambiguous with Self-Eval

5 Conclusion

In this article, we propose an LLM system with Self-Eval mechanism for SemEval-2024 Task 5. We explore the potential for using GPT-4 and prompt learning to obtain causal reasoning capabilities in the field of civil litigation. We have proven that the Self-Eval mechanism can alleviate the problem of unclear output and can also significantly improve performance. Additionally, we found that GPT4 demonstrates greater aptitude for choice tasks than for judgement tasks. With the prompts we provide, the experiment is fully reproducible and the experimental results can be extracted through regular expressions.

Due to time and space limitations, we leave some questions unresolved. For example, we only used GPT-4 for experiments. The broader applicability of the Self-Eval mechanism to other LLMs and its effectiveness in diverse tasks present room for further investigation. We intend to dive deeper into these questions in future work.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models. *CoRR*, abs/2309.10305.

IITK at SemEval-2024 Task 10: Who is the speaker? Improving Emotion Recognition and Flip Reasoning in Conversations via Speaker Embeddings

Shubham Patel* **Divyaksh Shukla*** **Ashutosh Modi**
Indian Institute of Technology Kanpur (IIT Kanpur)
{devang21, divyakshs23}@iitk.ac.in
{ashutoshm}@cse.iitk.ac.in

Abstract

This paper presents our approach for the SemEval-2024 Task 10: Emotion Discovery and Reasoning its Flip in Conversations. For the Emotion Recognition in Conversations (ERC) task, we utilize a masked-memory network along with speaker participation. We propose a transformer-based speaker-centric model for the Emotion Flip Reasoning (EFR) task. We also introduce *Probable Trigger Zone*, a region of the conversation that is more likely to contain the utterances causing the emotion to flip. For sub-task 3, the proposed approach achieves a 5.9 (F1 score) improvement over the task baseline. The ablation study results highlight the significance of various design choices in the proposed method.

1 Introduction

Conversations between participants carry information that evokes emotions. Emotions include personality, character, temper, and inspiration as the primary psychological parameters that drive them (P S and G S, 2017). Analyzing emotions through language helps uncover the interpersonal sentiments in a conversation at a finer level. This can help build better affective generative models (Goswamy et al., 2020), like chatbots that understand emotion and respond according to a person's behaviors and personality (Kumar et al., 2021; Colombo et al., 2019).

The SemEval-2024 Task 10 (Kumar et al., 2024) aims at Emotion Recognition (ERC), sub-task 1, and Emotion Flip Reasoning (EFR), sub-tasks 2 and 3, in conversations for two languages, namely English and Hindi-English Code-Mixed. ERC refers to identifying the emotion of different utterances. EFR is about identifying those utterances in the dialogue that caused the emotion of a speaker to change.

We build upon the models presented in Kumar et al. (2021) for ERC and EFR. A speaker's personality is likely to influence the emotions developed in other participants (Hazarika et al., 2018a). This inspired us to include information regarding speaker participation to improve the analysis of the emotion of an utterance in conversations. Additionally, for Emotion Flip Reasoning, we propose the Probable Trigger Zone (PTZ), a region of the conversation more likely to consist of the utterance that caused an emotional change in the target participant. This helps us filter out significant non-trigger utterances, reducing the skew in the data. We utilize pre-trained models for computing text embeddings to obtain better representations of utterances.

In sub-task 1, we achieved a weighted F1 score of 45 and 9th rank. For sub-tasks 2 and 3, we secured 5th and 10th position with F1 scores of 56 and 60, respectively. The top scores for each sub-task were 78, 79, and 79, respectively. For sub-task 3, our model improves 5.9 F1 over the baseline model presented in Kumar et al. (2021). The proposed changes have assisted in improving the performance of the system. A limitation of our model is knowing speakers. It might not be possible in all circumstances that this information is available. Also, despite trying to reduce the skew in the data, our model's performance was still impacted. Our models and code can be found here.[1]

2 Related Work

2.1 ERC

The task of emotion prediction has been of active interest in recent years (Witon et al., 2018; Kumar et al., 2020; Keswani et al., 2020; Singh et al., 2021b, 2023, 2021a), including the development of

* Equal Contributions

[1] https://github.com/Exploration-Lab/IITK-SemEval-2024-Task-10-Emotion-Flip

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1811–1820

models like ICON (Hazarika et al., 2018a), COGMEN (Joshi et al., 2022), Instruct-ERC (Lei et al., 2023) and the models by Kumar et al. (2021). Also, there has been active research in affective text generation (Goswamy et al., 2020). Several datasets exist (Bedi et al., 2023; Poria et al., 2019; Busso et al., 2008) that use one or more additional emotions along with Ekman's scheme (Ekman, 1992) of emotion representation via six emotion classes, namely, *fear*, *anger*, *joy*, *sad*, *disgust*, and *surprise*.

Hazarika et al. (2018a) and Li et al. (2020) highlight the importance of inter and intra-speaker interactions in a conversation. Li et al. (2020) achieves this by using three separate transformer-encoder blocks: (1) Conventional masking: masked multi-head self-attention, (2) Intra-speaker masking: all utterances from other speakers are masked, and (3) Inter-speaker masking: all utterances from the current speaker are masked. While this captures relationships, it does not capture the speaker's personality or presence. Hazarika et al. (2018a) also considers speakers, but it was modeled on the IEMOCAP dataset (Busso et al., 2008) that contains only two participants.

Shapes of Emotion (Bansal et al., 2022), ICON (Hazarika et al., 2018a) and its derived model ERC_MMN (Kumar et al., 2021) proposed the concept of speaker-level outputs, which means that during conversational flow, there is a speaker-level GRU to encode the currently spoken utterance. They achieve this by storing vectors representing each speaker and updating them using the speaker-level GRUs' hidden outputs, which are initialized to 0 during the start of a dialogue.

COGMEN (Joshi et al., 2022) introduces the concept of graphs to conversation flow for emotion recognition. They represent a graph in which each utterance is a node and is related to past or future utterances of the same or different speaker within a time window. CORECT (Nguyen et al., 2023) leverages on COGMEN and introduces speaker embeddings from MMGCN (Wei et al., 2019) to encode each speaker in the conversation for graph-based interaction and pairwise cross-modal feature interaction.

2.2 EFR

Kumar et al. (2021) introduces the relatively new Emotion-Flip Reasoning (EFR) task, which aims to identify past utterances in a conversation that have triggered one's emotional state to flip at a certain time. The task of Emotion-Cause Pair Extraction (ECPE) (Xia and Ding, 2019) and Emotion Cause Extraction (ECE) (Gui et al., 2016) are similar to EFR, but they aim to extract the causes of emotions from a given text instead of conversations. Kumar et al. (2021) present a transformer-based model for EFR and also measure the performance of baseline models CMN (Hazarika et al., 2018b), ICON (Hazarika et al., 2018a), DGCN (Ghosal et al., 2019), AGHMN (Jiao et al., 2019), and Pointer Network (Vinyals et al., 2017).

2.3 Embeddings

The performance of models on tasks is influenced by the quality of text representation it uses (Asudani et al., 2023). Nayak and Joshi (2022) release HingBERT, a BERT model that has been fine-tuned on Hindi-English Code-Mixed corpus. Muennighoff et al. (2023) introduce the Massive Text Embedding Benchmark (MTEB), which evaluates the performance of text embeddings through different tasks across several datasets. One of the top performers, the `voyage-embeddings`[2], utilize neural-net models to encode the text into text embeddings.

3 Task

SemEval-2024 Task 10: "**E**motion **Di**scovery and its **Re**asoning it **F**lip in Conversations" (Kumar et al., 2024), EDiReF, consisted of three sub-tasks:

1. ERC in Hindi-English Code-Mixed.
2. EFR in Hindi-English Code-Mixed.
3. EFR in English.

Emotion Recognition in Conversations (ERC) is classifying the utterances in a dialogue into one of the given emotion categories. An emotion flip is said to have occurred when a speaker's utterance differs from his/her previous utterance's emotion. Emotion Flip Reasoning (EFR) refers to identifying the utterances (triggers) that caused an emotional flip. These utterances could have been spoken either by the speaker himself or someone else. For the task of ERC, given the utterances in the conversation and corresponding speaker names, the emotion label for each utterance has to be predicted. For the task of EFR, the emotion labels of utterances have also been provided, and the triggers for a given emotion flip have to be predicted.

[2] https://docs.voyageai.com/embeddings/

Sub-task	1	2	3
Emotions	8	8	7
Episodes	343	452	833
Utterances	8506	11260	8747
Triggers	-	6542	5575

Table 1: Statistics of the Training Dataset.

Sub-Tasks 1 and 2 tailor the Hindi-English Code-Mixed dataset - MaSaC (Bedi et al., 2023). The training dataset consists of utterances in Roman script (e.g., "*yah plastic ke stickers tumne kahan se khariden?*"). Sub-Task 3 uses the MELD-FR dataset presented in Kumar et al. (2021) built upon the MELD dataset (Poria et al., 2019). Table 1 highlights the overall statistics regarding the training set for each sub-task. Figure 1a and Figure 1b show the distribution of the triggers as a function of the distance from the target utterance for the sub-task 2 and sub-task 3 training datasets, respectively.

4 System overview

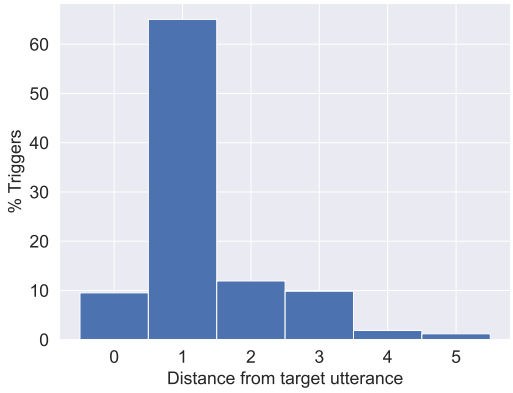
Inspired by the use of memory networks by Hazarika et al. (2018a) and Kumar et al. (2021) for emotion recognition, in 4.2 we present our model for the task of ERC, sub-task 1. Inspired by a transformer-based (Vaswani et al., 2017) approach for emotion flip reasoning presented by Kumar et al. (2021), in 4.3 we present our models for sub-tasks 2 and 3.

4.1 Utterance Embeddings

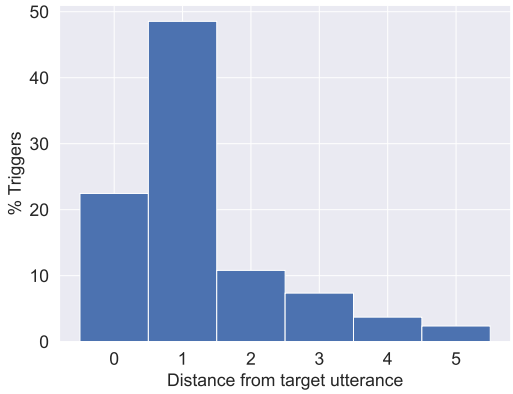
We utilize pre-trained models to compute representations of the utterances in the conversation. Sub-tasks 1 and 2 required the computation of utterance embeddings for code-mixed Hindi-English sentences. We utilized HingBERT to compute the utterance embedding as an average of all the token embeddings in an utterance. Sub-Task 3 consists of utterances in English. We referred to the Massive Text Embedding Benchmark to determine an efficient method to compute utterance embeddings. We experimented with the embeddings presented in Li and Li (2023) and voyage-embeddings, out of which the latter performed better. Hence, we used the `voyage-lite-02-instruct` model with `query_type` as a `document`.

4.2 ERC

We take inspiration from the Masked Memory Network architecture presented by Kumar et al. (2021)



(a) Sub-Task 2.



(b) Sub-Task 3.

Figure 1: Distribution of the distance between the target utterance and the causal utterance for emotion flip.

and speaker-specific GRUs proposed by Kumar et al. (2021) and Hazarika et al. (2018a). We used HingBERT to encode each utterance and then pass them through a dialog-level GRU and a speaker-level GRU. The vectors from the global-level GRUs are passed through a memory network through multiple hops (a cycle of reading from memory and writing back to memory is called a hop. The output is taken from the final memory read operation.) Then, attention is computed between the memory and speaker-level outputs while masking future utterances and concatenating with speaker-level outputs to compute conversation-level outputs. Finally, the obtained features are passed through a trainable linear layer for predicting emotion class. Figure 2 shows the model architecture.

Notations: u_t denotes the embedded utterance at time t in a dialogue, while $s_t^{(k)}$ denotes the k^{th} speaker embedding for utterance u_t. $attention(q, k, v)$ is the attention operator applied on query q, key k and value v. r is the number of

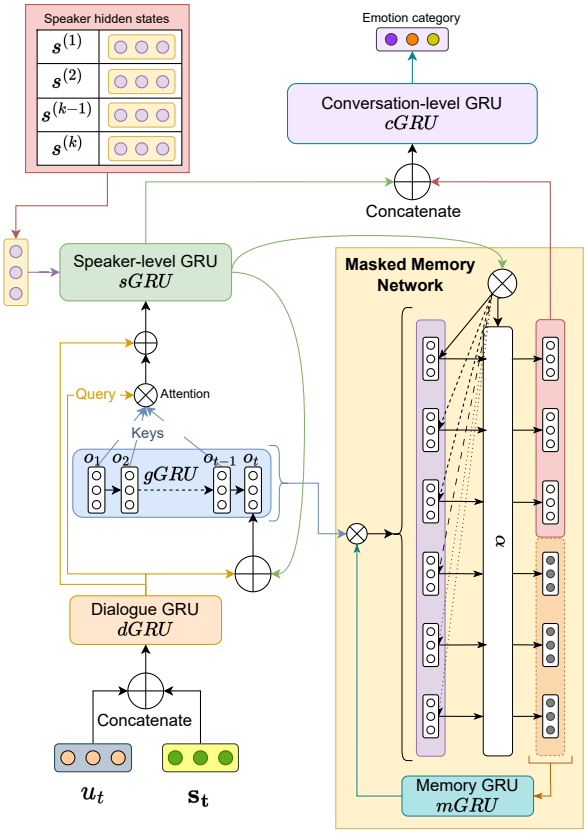


Figure 2: Masked Memory Network with Speaker-Embeddings concatenated with utterance embeddings. Speaker-embeddings are one-hot vectors of 6-dimensions which store 1 at the index of the top-6 speakers, otherwise 0.

hops in the memory layer, inspired from Hazarika et al. (2018a).

Dialogue-level GRU $dGRU$**:** This recurrent unit gives a dialogue-level representation of the u_t and gives output as do_t.

$$do_t = dGRU(u_t \oplus s_{kt})$$

Global-level GRU $gGRU$**:** This recurrent unit gives a global-level representation of the utterances $u_{(1:t)}$ till time step t, as $o_{(1:t)}$.

$$o_t = gGRU(do_t \oplus so_{t-1})$$

Attention Module: Attention is computed between do_t as query and value and $o_{(1:t-1)}$ as keys to obtain attention-based context for speaker-level GRU layer.

$$attention(do_t, o_{(1:t-1)}, do_t)$$

Speaker-level GRU $sGRU$**:** This gives a speaker-level recurrent unit that takes inputs $attention$ and speaker hidden state $s^{(k)}$ (taken from a dictionary of size k) and gives outputs so_t. The hidden output replaces the dictionary entry for $s^{(k)}$. At the start of a dialogue, the dictionary is empty, and the default hidden state for a new speaker is a zero vector.

$$so_t = sGRU(attention(do_t, o_{(1:t-1)}, do_t) + do_t)$$

Masked-Memory Attention: A memory vector, which represents the previous dialogues and utterances, is obtained by passing $o_{(1:t)}$ through a memory GRU ($mGRU$). This then goes through masked attention with the so_t while masking future utterances and a softmax activation α to give attention weights to each utterance in $o_{(1:t-1)}$. This is then used to update the memory vectors via the $mGRU$ and is concatenated with so_t as an input to $cGRU$.

$$temp = mGRU(o_{(1:t)})$$
$$mem^r = masked\_attention(temp, so_t)$$
$$temp = mGRU(mem^{r-1})$$

Conversation-level GRU $cGRU$**:** This layer represents the conversation flow of the dialogue and takes inputs as the concatenation of so_t and masked attention output, to give conversation-level features.

$$co_t = cGRU(mem^r + so_t)$$

Finally, the outputs of the $cGRU$ are used to compute the emotion class.

$$e_t = W.co_t + b$$

4.3 EFR

4.3.1 Baseline

In Kumar et al. (2021), the authors propose a transformer-based model for the task of EFR, whose architecture is as follows. Utterance embeddings for each utterance are computed using BERT (Devlin et al., 2019). The utterance embeddings of a conversation are passed through a transformer to compute a contextualized utterance embedding for each utterance. The emotion classes for each utterance are encoded as a one-hot vector and passed through a GRU to compute the emotion-history vector. For each utterance, its contextualized embedding, the contextualized embedding of the target utterance, and the emotion-history vector are passed through a linear layer to make a prediction.

4.3.2 Speaker-Aware Embeddings

As highlighted by Li et al. (2020) and Hazarika et al. (2018a), speaker interaction also drives the emotion of an utterance. Unlike their approach for modeling intra and inter-speaker interaction, we believe that the participation of certain speakers in the conversation drives the flip in the emotion of an utterance. Providing information regarding the speaker will help the model learn the nature of the specific speakers. To incorporate this, we utilize that the speakers in the test and the train set overlap.

An aspect of the conversation that has not been captured by the baseline model regarding the speakers of an utterance. In the baseline model, each utterance is treated as an independent text, and its embedding has been computed. This has failed to incorporate the information regarding who the speaker of a given utterance was. To incorporate this aspect, we concatenate the utterance embeddings with a one-hot vector denoting the speaker to create speaker-aware embeddings. This equips the model with the ability to capture the behavioral trends of specific speakers.

4.3.3 Probable Trigger Zone (PTZ)

We propose a hypothesis regarding the possible location of triggers. We divide the conversation into two parts. The first part consists of the utterances before the target speaker's previous utterance. The second part consists of utterances from the target speaker's previous utterance to his target utterance. We call the second part the Probable Trigger Zone (PTZ).

We hypothesize that no triggers lie in the first part of the conversation. Since the target speaker's emotion has flipped during the second part of the conversation, it is more likely that the causes for the emotion flip lie in the second part. Suppose the trigger causing the target emotion had been in the first part. In that case, it is more likely that it would have already caused the emotion of the previous utterance of the target speaker to flip. Then, the same emotion would have been carried to the target utterance, wrongly implying that no emotion flip occurred at the target utterance. To incorporate the hypothesis, we mask any predictions made by the model outside the Probable Trigger Zone. In section 5, we discuss how PTZ helps to reduce skew in the dataset.

For example, consider the conversation in Figure 3. Here, the target speaker is *Ross* with the target

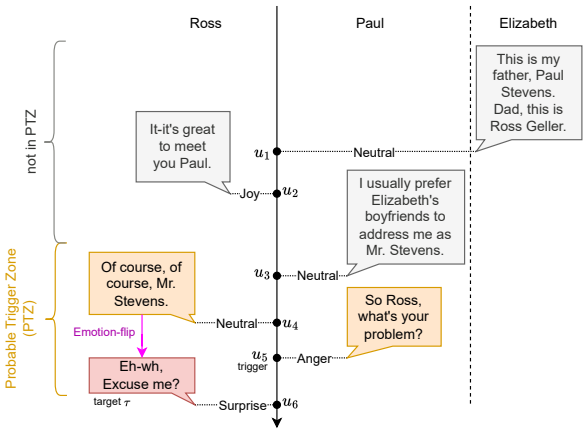


Figure 3: Probable Trigger Zone.

utterance u_6 and his previous utterance u_4. The probable trigger zone consists of utterances from u_4 to u_6. Due to the "surprise-causing" statement u_5 in PTZ, *Ross's* emotion flips from *Neutral* to *Surprise*. If this "surprise-causing" statement had been present outside the PTZ, i.e., before the previous utterance u_4, then the emotion of u_4 would likely have been *Surprise*.

4.3.4 Emotion-Aware Embeddings

Using an Emotion-GRU, the baseline model computes an emotion-history vector from the emotion labels. It uses this emotion-history vector in the final linear layer to predict the utterance label. A possible shortcoming of the above is that the linear layer has access only to the emotion history rather than to the emotion labels of the individual utterances. Also, the transformer layer cannot access the emotion labels while computing the contextualized utterance embeddings. Providing those to the transformer will also allow the embeddings to be emotion-aware. We concatenate our speaker-aware utterance embeddings and one-hot emotion labels to incorporate the above.

4.3.5 Model Functioning

Figure 4 presents the model architecture used for the task of EFR. The target utterance is denoted by the subscript τ. Each utterance u_t of a dialogue d is concatenated with its true emotion label e_t and one-hot speaker embedding s_t. This is then passed through the transformer to take into account the context. The Emotion-GRU computes the emotion-history vector. For each utterance, its and the target utterance's contextualized representation and the emotion-history vector are passed through a linear layer to make the prediction.

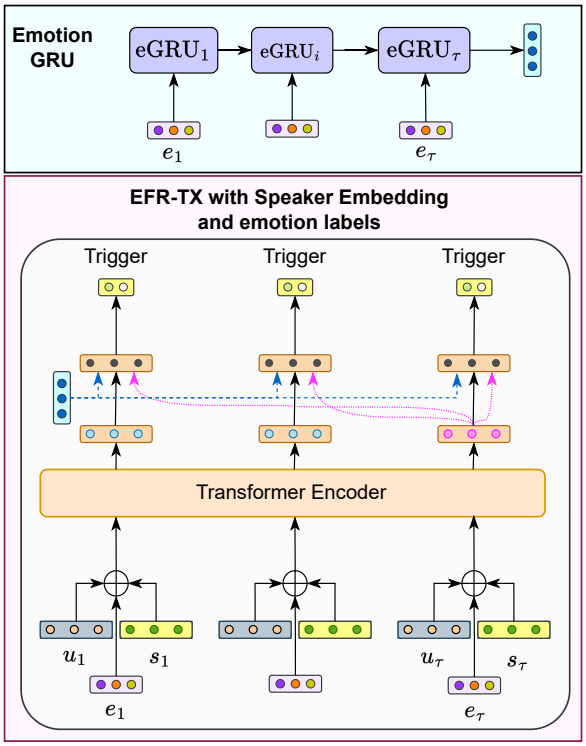


Figure 4: Architecture of the model proposed for the task of Emotion Flip Recognition.

5 Experimental setup

5.1 Training Details

For sub-task 1, we chose a sequence length seq_len of 15, i.e., we break a long conversation into disjoint smaller conversations with utterances less than equal to seq_len. For sub-tasks 2 and 3, we use a window size w of 5. We consider only the last w utterances in a conversation to predict the trigger, i.e., $U_{n-w+1}, U_{n-w+2}...U_{n-1}, U_n$. Table 2 contains details of the hyperparameters we used to train the models. To limit the size of the vector denoting the speaker, keep retained information regarding the top $k = 6$ speakers. We chose the top 6 speakers since this covered nearly $80 - 85\%$ of utterances in the corpus. We used Adam optimizer (Kingma and Ba, 2017) for all the sub-tasks, with a weight decay of 1e-5. Training of models has been done using Kaggle[3] P100 GPUs.

5.2 Effect of Hypothesis and Sequence Length

In Table 3 and Table 4, we highlight the impact of the hypothesis and selection of sequence length on the datasets. On reducing the window size w to 5, a significant number of negative labels have been eliminated, while there has not been much impact on the number of positive labels. Applying the hypothesis has helped mitigate the skew in the data, although there has been a slight impact on the number of positive labels. *Setting 1* refers to considering only the utterances within the window size $w = 5$. *Setting 2* refers to considering utterances that are both within the window and in the probable trigger zone.

[3] https://www.kaggle.com/

Sub-Task	1	2	3
Embedding Size	768	768	1024
Batch Size	64	2000	1000
Learning Rate	1e-04	5e-07	5e-07
Weights	Inv Sqrt	Inv	Inv
Epochs	100	1000	1000
Best Epoch	80	299	549
Training Time	10 hr	3 hr	3 hr

Table 2: Hyperparameters for each of the sub-tasks. Weights refers to the weights in the cross entropy loss. *Inv*: Inverse of the supports. *Inv Sqrt*: Inverse of the square root of the supports.

Dataset	0	1	Ratio
Original	92233	6544	14.1
Setting 1	17539	6425	2.7
Setting 2	11535	5839	2.0

Table 3: Effect of PTZ on Dataset, Sub-Task 2. *Setting 1* and *Setting 2* as defined in Section 5.2.

Dataset	0	1	Ratio
Original	29416	5575	5.3
Setting 1	13483	5177	2.6
Setting 2	7834	4542	1.7

Table 4: Effect of PTZ on Dataset, Sub-Task 3. *Setting 1* and *Setting 2* as defined in Section 5.2.

6 Results

For Sub-Task 1, the dataset consisted of a non-uniform distribution of labels, with *neutral* being the most frequent. A model that predicts the emotion category of each utterance to be *neutral* achieves a weighted F1 of 24.36. We consider this as a simple *neutral* baseline for sub-task 1. For Sub-Task 2, we have kept the baseline as a rule-based model that predicts the previous utterance to be a trigger and the rest of all utterances non-triggers. The data for the second sub-task is highly skewed

as can be seen in Figure 1a. Due to this baseline performs exceptionally well, as can be observed in Table 5. For Sub-Task 3, we use ERC$^{\text{True}}$ EFR-TX from Kumar et al. (2021) as the baseline.

Sub-Task	Model	Metric	Value
2	Rule-Based	F1	79.15
3	ERC$^{\text{True}}$ EFR-TX	F1	53.9

Table 5: Baselines for various Sub-Tasks.
Rule-Based: A rule-based model that predicts the previous utterance as a trigger and the rest as non-triggers.

6.1 Model Performance

We have highlighted the performance of our models on the test data in Table 6. For sub-task 2, we get precision and recall scores of 0.73 and 0.83, respectively. For sub-task 3, we get precision and recall scores of 0.74 and 0.80, respectively.

6.2 Error Analysis

For sub-task 1 and sub-task 3, our model performed better than the baselines, but not for sub-task 2. For sub-task 1, the dataset consisted of a non-uniform distribution of labels in the training dataset. Due to this skew in the data, the model has shown different performances for different labels. The predictions for sub-task 1 have been highlighted in Figure 5. For EFR, the usage of a window size $w = 5$ utterances has helped to eliminate a large number of non-triggers. Due to this, the model's predictions have many true negatives, as can be seen in Table 7 and Table 8. But despite this, there was still skew in the data, which impacted the model's performance in predicting the minority class of triggers. The data for sub-task 2 is highly skewed, as can be seen in Figure 1a. We suspect this is why our model has performed poorer than the baseline.

Figure 6 is an example of an erroneous emotion labeling of the model on the test set of sub-task 1 (ERC). Here, the utterance marked in red has the true label as 'Fear,' but the model predicts 'Neutral.' This is due to the sharp change in conversation context at u_5. Also, 'hahhaha' directly corresponds to laughing, but in this case, the speaker at u_6 utters 'hahhaha' as he is worried that the inspector is looking for 'Sharman'. The speaker, Indravardhan, who continuously interacted with the inspector, showed neutral emotion. This led to storing vectors corresponding to neutral for Indravardhan in the memory network, leading to misclassification of emotion at u_6.

Sub-Task	Metric	Our	Best	Rank
1	Weighted F1	44.80	78	9
2	F1	56.35	79	5
3	F1	59.78	79	10

Table 6: Results on the Test Set.

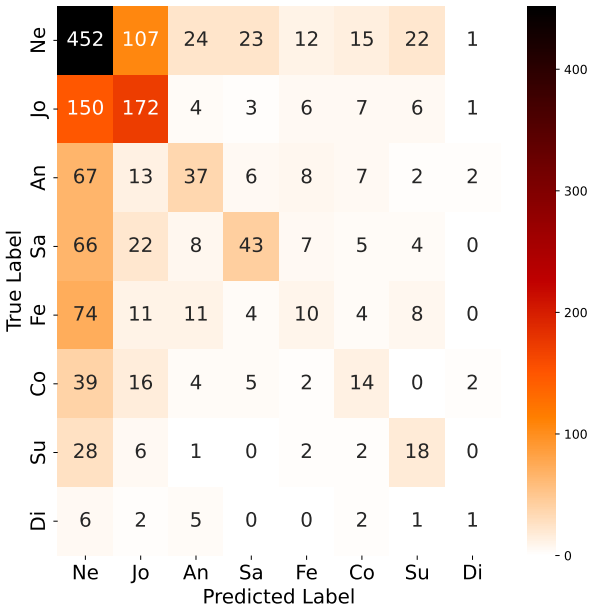


Figure 5: Confusion Matrix for Sub-Task 1.

6.3 Ablation Study

The application of the hypothesis has assisted in removing a few of the wrongly guessed triggers. This has improved the model's performance, as seen in Table 9. We also experimented with another approach of making predictions only inside the PTZ instead of masking the outside labels. This was done by training in the model and making predictions only using the utterances within the probable target zone. In this case, the model's performance was poorer. We suspect this is because the context the model gets in the latter is more restricted than in the first case. Due to this, the model is not able to make predictions effectively.

7 Conclusion

In this paper, we discussed approaches to improve the masked memory network architecture for emotion recognition (ERC) and transformer-based architecture for emotion flip reasoning (EFR) by incorporating speaker information into the embeddings and making better use of the emotion labels. We also employed an approach of focusing the

		Predicted	
		0	**1**
True	**0**	6943	331
	1	123	293

Table 7: Confusion Matrix Sub-Task 2.

		Predicted	
		0	**1**
True	**0**	6735	738
	1	356	813

Table 8: Confusion Matrix Sub-Task 3.

model's prediction on more likely regions to identify triggers by defining the Probable Trigger Zone in conversations. This, along with considering a window of last-few utterances, assisted in reducing the bias in the data.

Limitations: Our model assumes that the training and testing data consist of the same speakers. While this would be true for many benchmark datasets of emotion analysis in conversations, it might not be true in all real-world circumstances. Another limitation of the proposed approach is the training time.

Future Work: In the future, we can include speaker information across dialogues for ERC to capture better speaker semantics by using learnable embeddings for each speaker updated by the hidden outputs of the speaker-level GRU. However, to apply the above, we need to know the number of speakers in the datasets, training, and testing. Additionally, the model becomes dependent on the number of speakers.

A possible approach to help mitigate the assumption of having common speakers and knowing the number of speakers in the training and test time could be exploring further modeling inter and intra-speaker dependencies, as shown in Li et al. (2020) and Hazarika et al. (2018a). They propose models that capture speaker relationships but are not dependent on the number of speakers.

Mitigating the issues of skewed data can be further explored to enhance the system's performance. Also, addressing other aspects of conversations, such as whom the statement is being told to and treating names of speakers in utterances differently from simple pronouns, can be explored.

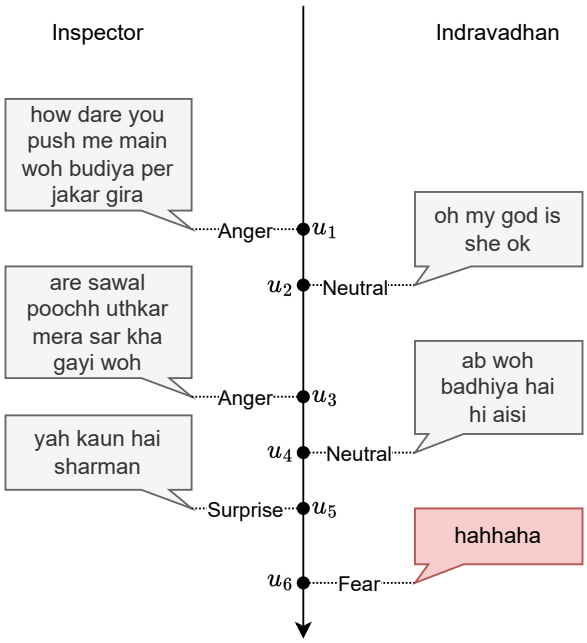


Figure 6: Example of an erroneous emotion labeling from the model. The true label is 'Fear,' but the model predicted 'Neutral.'

Sub-Task	Masks	F1 Before	F1 After	Change
2	1	56.29	56.35	+0.06
3	78	58.68	59.78	+1.10

Table 9: Improvements by PTZ.
Masks: The number of trigger predictions made by the model outside the Probable Trigger Zone, which had been masked to 0.

References

Deepak Suresh Asudani, Naresh Kumar Nagwani, and Pradeep Singh. 2023. Impact of word embedding models on text analytics in deep learning environment: a review. *Artif. Intell. Rev.*, 56(9):10345–10425.

Keshav Bansal, Harsh Agarwal, Abhinav Joshi, and Ashutosh Modi. 2022. Shapes of emotions: Multimodal emotion recognition in conversations via emotion shifts. In *Proceedings of the First Workshop on Performance and Interpretability Evaluations of Multimodal, Multipurpose, Massive-Scale Models*, pages 44–56, Virtual. International Conference on Computational Linguistics.

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Transactions on Affective Computing*, 14(2):1363–1375.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S.

Narayanan. 2008. IEMOCAP: interactive emotional dyadic motion capture database. *Lang. Resources &. Evaluation*, 42(4):335–359.

Pierre Colombo, Wojciech Witon, Ashutosh Modi, James Kennedy, and Mubbasir Kapadia. 2019. Affect-driven dialog generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3734–3743, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 154–164, Hong Kong, China. Association for Computational Linguistics.

Tushar Goswamy, Ishika Singh, Ahsan Barkati, and Ashutosh Modi. 2020. Adapting a language model for controlled affective text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2787–2801, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, and Yu Zhou. 2016. Event-driven emotion cause extraction with corpus construction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1639–1649, Austin, Texas. Association for Computational Linguistics.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018a. ICON: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604, Brussels, Belgium. Association for Computational Linguistics.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018b. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2122–2132, New Orleans, Louisiana. Association for Computational Linguistics.

Wenxiang Jiao, Michael R. Lyu, and Irwin King. 2019. Real-time emotion recognition via attention gated hierarchical memory network.

Abhinav Joshi, Ashwani Bhat, Ayush Jain, Atin Singh, and Ashutosh Modi. 2022. COGMEN: COntextualized GNN based multimodal emotion recognitioN. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4148–4164, Seattle, United States. Association for Computational Linguistics.

Vishal Keswani, Sakshi Singh, Suryansh Agarwal, and Ashutosh Modi. 2020. IITK at SemEval-2020 task 8: Unimodal and bimodal sentiment analysis of Internet memes. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1135–1140, Barcelona (online). International Committee for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Ayush Kumar, Harsh Agarwal, Keshav Bansal, and Ashutosh Modi. 2020. BAKSA at SemEval-2020 task 9: Bolstering CNN with self-attention for sentiment analysis of code mixed text. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1221–1226, Barcelona (online). International Committee for Computational Linguistics.

Shivani Kumar, Md Shad Akhtar, Erik Cambria, and Tanmoy Chakraborty. 2024. Semeval 2024 – task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md. Shad Akhtar, and Tanmoy Chakraborty. 2021. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *CoRR*, abs/2103.12360.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework. *arXiv preprint arXiv:2309.11911*.

Jiangnan Li, Zheng Lin, Peng Fu, Qingyi Si, and Weiping Wang. 2020. A hierarchical transformer with speaker modeling for emotion recognition in conversation.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Cam Van Thi Nguyen, Tuan Mai, Son The, Dang Kieu, and Duc-Trong Le. 2023. Conversation understanding using relational temporal graph neural networks with auxiliary cross-modality interaction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

SREEJA P S and Mahalakshmi G S. 2017. Emotion models: A review. *International Journal of Control Theory and Applications*, 10:651–657.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy. Association for Computational Linguistics.

Aaditya Singh, Shreeshail Hingane, Saim Wani, and Ashutosh Modi. 2021a. An end-to-end network for emotion-cause pair extraction. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 84–91, Online. Association for Computational Linguistics.

G. Singh, D. Brahma, P. Rai, and A. Modi. 2021b. Fine-grained emotion prediction by modeling emotion definitions. In *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–8, Los Alamitos, CA, USA. IEEE Computer Society.

Gargi Singh, Dhanajit Brahma, Piyush Rai, and Ashutosh Modi. 2023. Text-based fine-grained emotion prediction. *IEEE Transactions on Affective Computing*, pages 12–12.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2017. Pointer networks.

Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1437–1445, New York, NY, USA. Association for Computing Machinery.

Wojciech Witon, Pierre Colombo, Ashutosh Modi, and Mubbasir Kapadia. 2018. Disney at IEST 2018: Predicting emotions using an ensemble. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 248–253, Brussels, Belgium. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

DeepPavlov at SemEval-2024 Task 8: Leveraging Transfer Learning for Detecting Boundaries of Machine-Generated Texts

Anastasia Voznyuk and **Vasily Konovalov**
Moscow Institute of Physics and Technology
{vozniuk.ae, vasily.konovalov}@phystech.edu

Abstract

The Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection shared task in the SemEval-2024 competition aims to tackle the problem of misusing collaborative human-AI writing. Although there are a lot of existing detectors of AI content, they are often designed to give a binary answer and thus may not be suitable for more nuanced problem of finding the boundaries between human-written and machine-generated texts, while hybrid human-AI writing becomes more and more popular. In this paper, we address the boundary detection problem. Particularly, we present a pipeline for augmenting data for supervised fine-tuning of DeBERTaV3. We receive new best MAE score, according to the leaderboard of the competition, with this pipeline.

1 Introduction

Recently, there has been a rapid development of auto-regressive language models, for example, GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023), and LLaMA2 (Touvron et al., 2023). These models are trained on enormous amounts of data and are able to produce coherent texts that can be indistinguishable from human-written texts (Dugan et al., 2022).

The SemEval-2024 Task 8 competition suggests to tackle the problem of detecting machine-generated texts. This problem has become more relevant recently due to the release of ChatGPT[1], a model by OpenAI that simplified the access to the large language models (LLM) and their usage. For example, LLM can be maliciously used to generate fake news (Zellers et al., 2019). There are also some concerns raised among scientists (Ma et al., 2023) and educators (Zeng et al., 2023) that the usage of LLMs will devalue the process of learning and research.

The commonly used approach to formulate the task of detecting machine-generated texts is a binary classification task (Jawahar et al., 2020). In this case, a text can be attributed to either a human or a LLM. Otherwise, the task can be formulated as a multiclass classification or an authorship attribution task (Uchendu et al., 2020), where it is needed to determine which one of the k authors is the real author of the given text. Finally, the trend toward human-AI collaborative writing is rising, which highlights the importance of the boundary detection task. In this setup, text contains consecutive chunks of different authorship, and it is required to detect where the boundaries between chunks lie and who is the author of every chunk. Due to its complexity, it is usually assumed the text has a human-written prefix and the rest of the text is AI-generated (Dugan et al., 2022; Cutler et al., 2021; Kushnareva et al., 2023).
Our main contributions are three-fold:

1. We receive the new best MAE score on the task of detecting the boundary between human-written and machine-generated parts of the text.

2. We present a new simple yet effective pipeline of augmenting data for the task of boundary detection, which allows us to get more data for training and improve the results of fine-tuning large language models.

3. We compare the performance of several fine-tuned models with different architectures on various amounts of training data.

Additionally, we've made the code of augmentation publicly available.[2]

[1] https://openai.com/blog/chatgpt

[2] https://github.com/natriistorm/semeval2024-boundary-detection

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1821–1829

2 Related Work

Most of approaches (Jawahar et al., 2020) for machine-generated text detection are based on calculating linguistic (Fröhling and Zubiaga, 2021), stylometric, and statistical features, as well as on using classical machine learning methods like logistic regression, random forest, and gradient boosting as classifiers. Among commonly used features are word and n-gram frequencies (Manjavacas et al., 2017), and tf-idf (Solaiman et al., 2019).

An alternative strategy is to use zero-shot techniques based on the internal metrics of the texts. For example, token-wise log probability can be evaluated by models like GROVER (Zellers et al., 2019) or GPT-2 (Solaiman et al., 2019). A probability threshold is established to distinguish writings produced by machines from those written by humans. Moreover, rank (Gehrmann et al., 2019) or log-rank (Mitchell et al., 2023) can be calculated for each token and compared for consistency with the prior context.

It's shown by Ippolito et al. (2020) that feature-based methods are inferior to methods based on using encoders of pretrained language models like BERT (Devlin et al., 2019) as a basis for fine-tuning on the selected domain. Representations from auto-regressive language models can be used as input for the classification head. Such transformer-based methods require supervised detection examples for further training. Among the models commonly used for fine-tuning are RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019). Fine-tuned RoBERTa has shown the state-of-the-art results on the problems of authorship attribution (Uchendu et al., 2021).

To tackle the problem of mixed human-machine writing, Dugan et al. (2022) introduces the Real Or Fake Text (RoFT) tool, where humans were asked to detect the sentence where the text transitions from human-written text to machine-generated text. One of the possible formulations of this task is multilabel classification (Cutler et al., 2021), where the boundary detector needs to determine the first generated sentence, and the number of this sentence is considered the label of the text. In that work, each sentence is processed separately with shallow classification and regression models based on RoBERTa and SRoBERTa (Reimers and Gurevych, 2019). That solution perform well in an in-domain setup, but is limited in an out-of-domain setup.

Any solution for tasks about detecting machine-generated texts should be robust to domain change. The organisers of the SemEval-2024 Task 8 competition claimed to have added new domains to the test set for testing the robustness of participants' solutions. There are several works on performance of detecting methods on out-of-domain setup, such as Kushnareva et al. (2023) and Zeng et al. (2023). Kushnareva et al. (2023) conclude that perplexity-based and topological features appear to be helpful in case of domain shift.

3 Data and Task Description

3.1 Task Description

The Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection (Wang et al., 2024) is focused on challenging detectors of machine-generated texts. The dataset, provided by organisers, consists of 3 parts:

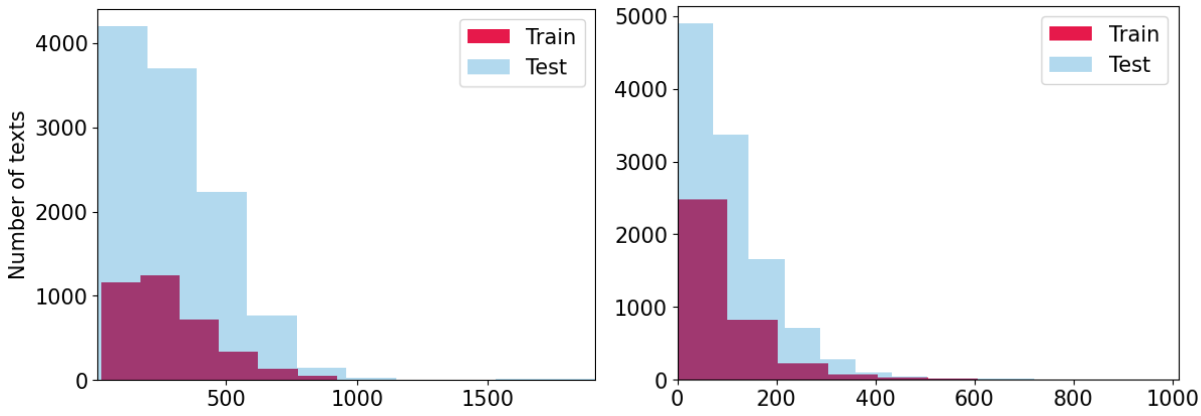
1. texts of different authorship for subtask A and subtask B;

2. texts with collaborative human-AI writing for subtask C.

This paper focuses only on subtask C, suggesting a solution to differentiate a human-written prefix from the rest of the AI-generated text. The texts for this subtask are generated in the following way: the language model should continue the human-written text, which is given as a prompt. Several examples of texts are presented in Appendix A. The designated evaluation metric for this task is Mean Absolute Error (MAE), which quantifies the absolute distance between the predicted word and the actual word where the switch of authorship between human and LLM occurs.

3.2 Data Description

The dataset for this task is derived from the M4 (Wang et al., 2023) dataset, which contains texts of various domains, various languages, and generators. The authors show that current detectors tend to misclassify machine-generated texts as human-written if they're given a text from a different domain. The texts in the train and validation datasets are generated from scientific paper reviews from PeerRead (Kang et al., 2018). The test set partially consists of texts generated from PeerRead. In order to check the robustness of the solutions to domain shift, texts from Outfox, a dataset of LLM-generated student essays (Koike et al., 2023), are



(a) Distribution of lengths of tokenized texts

(b) Distribution of boundary subtoken positions

Figure 1: Statistics of the texts in the datasets

added to the dataset. All data sets contain English texts only.

3.3 Data Analysis

The distribution of lengths of texts in train and test data sets, tokenized by DeBERTa-V3-base models, is shown in Figure 1a. The majority of the texts in the test set are shorter in length, but there are several texts that exceed 1,000 subtokens. The distribution of positions of a boundary subtoken in texts is shown in Figure 1b. A boundary subtoken is the first subtoken of a boundary word. In both datasets, there is a distinguishable disproportion in the position of a boundary subtoken, as most of them have the boundary subtoken within the first 200 subtokens. Thus, a model, trained only on this data, will give limited results when encountering longer texts with longer human prefixes in them.

4 Proposed Method

The provided train dataset consists only of 3,649 texts. It's well known that an abundance of in-domain training data is crucial for classifier performance (Konovalov et al., 2016). However, during the competition period, it was prohibited to use any external data for training, and thus we were limited to working only with the provided dataset. In this case, getting more training data with some kind of augmentation plays a crucial role. We designed an augmentation pipeline and ran all our experiments on two sets of data: the one provided by the organisers of the competition, described in Section 3, and our augmented data.

4.1 Data Augmentation

The general idea of our augmentation pipeline is to split the text into distinct sentences and take several consequent sentences from the text with authorship change. It will make new texts coherent and will not mislead models during training (Ostyakova et al., 2023). In addition, to be useful for training, each sequence should contain a sentence with an authorship change.

Another nuance about the augmentation process is the need to correctly determine the boundary label. The boundary label is calculated as a number of whitespace-split words in the human-written prefix. However, the initial dataset contains texts where a pair or even a sequence of words is split only by line breaks or punctuation symbols. Such a sequence of words should be considered as one word when calculating the boundary label. Thus, we have to pay a lot of attention to whitespace characters during augmentation and do not mistakenly append new whitespace characters between words.

We preprocess each text in the dataset for augmentation in the following way:

1. Split the text into sentences by punctuation symbols.
2. Split the sentences themselves by whitespaces into lists of whitespace-split words.
3. Compare the list of whitespace-split words from previous step with the list of whitespace-split words obtained by splitting the text itself. In case of discrepancy, fix it, depending on its type.

See example of preprocessing for the text from train set in Appendix D.

In the third step of preprocessing, there are two main types of discrepancies: lost whitespace characters and words sequences, originally separated by line breaks only that were split during sentence split process. The former is solved by inserting the missing whitespace characters, while for the latter we merge the split words into one sequence. The last step of preprocessing is crucial and skipping it will result in the incorrect calculation of the boundary label for augmented text, as the label directly depends on the number of whitespace characters in the text.

After preprocessing, we take a number of consecutive sentences to the left and to the right from the boundary sentence, combine them in a text and determine the label of the boundary word in this new text.

4.2 Model Comparison

We used only the transfer learning approach, where a pretrained transformer-based model is fine-tuned on our task in a supervised way. We would like the model to be able to work with long enough sequences because we want the whole human-written prefix to fit in the encoder. Thus, we've determined three models that showed good results on the task of machine-generated text detection:

1. RoBERTa (Liu et al., 2019) has shown good performance in both tasks of boundary detection (Kushnareva et al., 2023) and machine-generated text detection (Macko et al., 2023).

2. Longformer (Beltagy et al., 2020), which was suggested as a baseline by organisers of the task. This model is based on pretrained RoBERTa with novel attention mechanism with a sliding window to long sequences.

3. DeBERTa (He et al., 2021b), which is the state-of-the-art model for machine-generated text detection (Macko et al., 2023). It overcomes the BERT and RoBERTa by introducing a disentangled attention for encoding the position and content of each token separately into two vectors. We decided to test it in the boundary detection task to understand whether it outperform RoBERTa. In our experiments, we fine-tuned DeBERTaV3 (He et al., 2021a) which is an enhanced version of DeBERTa.

All three models are fine-tuned for token classification task. For each token, models predict the probability of being a boundary token and output the most probable token.

Model	dev	test
RoBERTa-base	9.04 \ 5.78	31.56 \ 30.71
RoBERTa-large	6.72 \ 4.18	25.25 \ 20.66
longformer-base	5.10 \ 5.67	23.16 \ 22.94
longformer-large	4.54 \ 4.40	22.97 \ 20.33
DeBERTaV3-base	3.66 \ 3.15	16.12 \ 13.98
DeBERTaV3-large	2.38 \ 2.54	15.16 \ **13.38**
Top-1 Submission	-	15.68

Table 1: MAE on original \ augmented dataset and comparison with Top-1 submission on the leaderboard. Longformer-base is suggested as a baseline solution by organisers of competition.

5 Experimental Setup

We have used pretrained `longformer-4096-base` and `longformer-4096-large` with default hyperparameters to fine-tune Longformer. For experiments with RoBERTa, we have chosen two models: `roberta-base` and `roberta-large`. The models were fine-tuned with the set of custom hyperparameters, taken from the original paper (Liu et al., 2019). Finally, for experiments with DeBERTa, we have also chosen two models: `deberta-v3-base` and `deberta-v3-large`. The models were fine-tuned with the set of custom hyperparameters taken from He et al. (2021a). All custom hyperparameters are listed in Appendix B.

For all models we set the maximum length of context in tokenizer equal to 512 as there are only few text items in both train and test set with tokenized text length greater than 512. Additionaly, we've used the early stopping method for all of our experiments to get rid of epochs dependency.

6 Results and Discussion

6.1 Main Results

In Tables 1, we compare MAE scores of different models from Section 4.2. There are experiments with models fine-tuned on the original dataset provided by organisers and on the extended dataset with both augmented and original texts.

All models perform better when they are fine-tuned on the extended dataset rather than only on original texts. It clearly shows that even such a simple data augmentation provides a significant boost in performance.

If we compare the results among the models, we will clearly see the dominance of DeBERTaV3 models. For both setups DeBERTaV3-large has shown the best performance and the lowest MAE score on the validation and test datasets. On the setup with the extended dataset, DeBERTaV3-large gets new best MAE score for the competition, which is equal to 13.375. It improves MAE score of the top-1 submission by more than 2 points, from 15.683 to 13.38.

6.2 Discussion

Results in Section 6.1 show the importance of both the variety of the data in the dataset and a pretrained model. Leveraging augmented training data significantly increases performance on the task, because it introduces variety in lengths of texts and boundary token positions while preserving the coherence of the texts. We believe that to be the reason why the models perform better when they are fine-tuned on the dataset with augmented data rather than on the original dataset.

Apart from various data, it is also important for the pretrained model to have great generalization capabilities. DeBERTaV3-large has better generalization capabilities than other tested models (He et al., 2021b). The advanced architecture of DeBERTaV3 helps to significantly improve the MAE score in comparison with both RoBERTa and Longformer.

For all three models, large vesion of each model outperforms base version on both the original and the extended dataset. The reason to this is, greater number of trainable parameters allows models to generalize on training set better.

6.3 Error Analysis

We manually inspected texts from the test set on which the best-performing model, DeBERTaV3-large, made serious mistakes. We've limited our inspection set to the texts where the distance between the predicted label and the true label was more than 100 tokens. Thus, we got 276 texts. Only 75 out of these 276 texts were from PeerReview domain, so model made most of its mistakes on the Outfox domain, texts from which are not present in the train set.

These two domains presented in the test set are very different: they vary in the style of formatting, punctuation, and text structuring. The second domain of LLM-generated student essays have a lot of spelling and punctuation problems and it may confuse the models, as they trained on more formal and literate texts. It would be interesting to evaluate models on each of these domains separately. However, because we do not have domain classification lables in the test set, it is not yet feasible.

6.4 Anomalies in Texts

In the majority of texts from the original dataset, the model generates a coherent continuation of the human-written prefix, and it may be hard for a human to guess the boundary word without knowing it. However, there are a number of texts in the data sets that have some flaws in the generated parts.

There are texts in which LLM hallucinated. It either repeated the human prefix or went into a loop where it generated excessive lists with the same beginning. See example of it in Appendix A and in Appendix C. Such hallucinations can be an immediate hint for detector model to put the label boundary near this anomaly. Also, sometimes machine-generated text can have some distinguished features that imply the artificial nature of a particular part of the text. For example, in a number of cases the model begins the generation with the """ (three double quotation marks). It may also be a hint for the detector. A list of other common features we've encountered while examining the test set is provided in Appendix C.

7 Conclusion

In this paper we describe the system submitted for SemEval2024-Task 8, the subtask dedicated to hybrid human-machine writing detection. We present a simple yet effective augmentation pipeline. We explore how adding this pipeline to the process of fine-tuning can significantly increase the performance on the task, and provide an analysis of performance of various models with and without our augmentation pipeline. The best model, which is DeBERTaV3-large fine-tuned on a large set of augmented data, receives a new best score according to the leaderboard of the competition. Other fine-tuned models achieve competitive results, ranking in the upper half of the leaderboard and beating the organisers' baseline. As the provided data was limited to English language only, future work might include training multilingual boundary detection solution by mixing training data of different languages and using a multilingual encoder (Chizhikova et al., 2023). Such a system can be used for hybrid AI-writing detection as a standalone solution or can

be integrated into existing NLP frameworks like DeepPavlov (Burtsev et al., 2018).

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, and Vasily Konovalov. 2018. Deeppavlov: An open source library for conversational ai. In *NIPS*.

Anastasia Chizhikova, Vasily Konovalov, and Mikhail Burtsev. 2023. Multilingual case-insensitive named entity recognition. In *Advances in Neural Computation, Machine Learning, and Cognitive Research VI*, pages 448–454, Cham. Springer International Publishing.

Joseph W. Cutler, Liam Dugan, Shreya Havaldar, and Adam Stein. 2021. Automatic detection of hybrid human-machine text boundaries.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2022. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *AAAI Conference on Artificial Intelligence*.

Vasily Konovalov, Oren Melamud, Ron Artstein, and Ido Dagan. 2016. Collecting Better Training Data using Biased Agent Policies in Negotiation Dialogues. In *Proceedings of WOCHAT, the Second Workshop on Chatbots and Conversational Agent Technologies*, Los Angeles. Zerotype.

Laida Kushnareva, Tatiana Gaintseva, German Magai, Serguei Barannikov, Dmitry Abulkhanov, Kristian Kuznetsov, Irina Piontkovskaya, and Sergey Nikolenko. 2023. Artificial text boundary detection with topological data analysis and sliding window techniques.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. 2023. Ai vs. human – differentiation analysis of scientific content generation.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Enrique Manjavacas, Jeroen De Gussem, Walter Daelemans, and Mike Kestemont. 2017. Assessing the stylistic properties of neurally generated text in authorship attribution. In *Proceedings of the Workshop on Stylistic Variation*, pages 116–125, Copenhagen, Denmark. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23.

OpenAI. 2023. Gpt-4 technical report.

Lidiia Ostyakova, Veronika Smilga, Kseniia Petukhova, Maria Molchanova, and Daniel Kornev. 2023. ChatGPT vs. crowdsourcing vs. experts: Annotating open-domain conversations with speech functions. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 242–254, Prague, Czechia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. *Defending against neural fake news*. Curran Associates Inc., Red Hook, NY, USA.

Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guanliang Chen. 2023. Towards automatic boundary detection for human-ai collaborative hybrid essay in education.

A Examples of Texts

Table 2 contains three examples of how authorship change occurs in the texts from the train set. While

the first text contains no signs of a flawed generation, the second and third texts have some flaws. In the second text, the model begins to generate from the capital letter, and its generation is incoherent with the human-written prefix. In the third text, the model starts to repeat the end of the human part, which is a glaring sign of machine-generated text.

Model
I noticed that in Figure 2, the two quantization factors for quantized layers are missing labels. It would be helpful for the reader to understand which layers are being quantized in the figure
Hi Authors, You seem to have submitted two of the same paper? Pls advise Could you please clarify if this is a mistake or if there are any differences between the two submitted papers?
There has been prior work on semi-supervised GAN, though this paper is the first context conditional variant. The novelty of the approach was the novelty of the approach was leveraging in-painting using an adversarial loss to generate contextually relevant images.

Table 2: Examples of texts from train set with different quality of LLM generation and with highlighted human prefix

B Hyperparameters

For fine-tuning DeBERTaV3 we use hyperparameters, listed by model authors in He et al. (2021a). Table 3 lists these hyperparameters.

Hyperparameters	Large	Base
Optimizer	AdamW	AdamW
Adam β_1, β_2	0.9, 0.999	0.9, 0.999
Adam ϵ	1e-6	1e-6
Warm-up step	50	50
Batch size	4	32
Learning rate (LR)	5e-6	2e-6
Learning rate decay	Linear	Linear
Weight decay	0.01	0.01
Gradient clipping	1.0	1.0

Table 3: Hyperparameters for fine-tuning DebertaV3-large and DebertaV3-base

Table 4 contains hyperparameters for fine-tuning RoBERTa, also taken from original paper (Liu et al., 2019).

Hyperparameters	values
Optimizer	AdamW
Warm-up steps	50
Batch size	16
Learning rate (LR)	5e-6
Learning rate decay	Linear
Weight decay	0.01

Table 4: Hyperparameters for fine-tuning RoBERTa

C Examples For Error Analysis

See Table 5.

D Augmenting Pipeline Scheme

See Figure 2.

Anomaly	Text Id	Example
Excessive repetition	613	...praying for. No more traffic jams, no more parking nightmares, no more car payments, no more insurance, no more maintenance, no more oil changes, no more tire rotations...
Extremely long lists of items	8854	...nice approach to """ + learning skills + learning skills in a sample efficient way + learning skills in an interpretable way + learning skills that can be used on downstream tasks + learning skills that are transferable between domains
JSON-structured hallucinations	5358	...Summary of revisions: * * * """, "title": "Diet Networks: Thin Parameters for Fat Genomics", "abstract": "Learning tasks such as...
Bizarre formatting	5639	...Below are my comments: (1) """ The first sentence of the abstract is too long. It should be divided into two sentences (2) """

Table 5: Table with some frequent anomalies in the generated part of texts from test set. The highlighted part is human-written prefix.

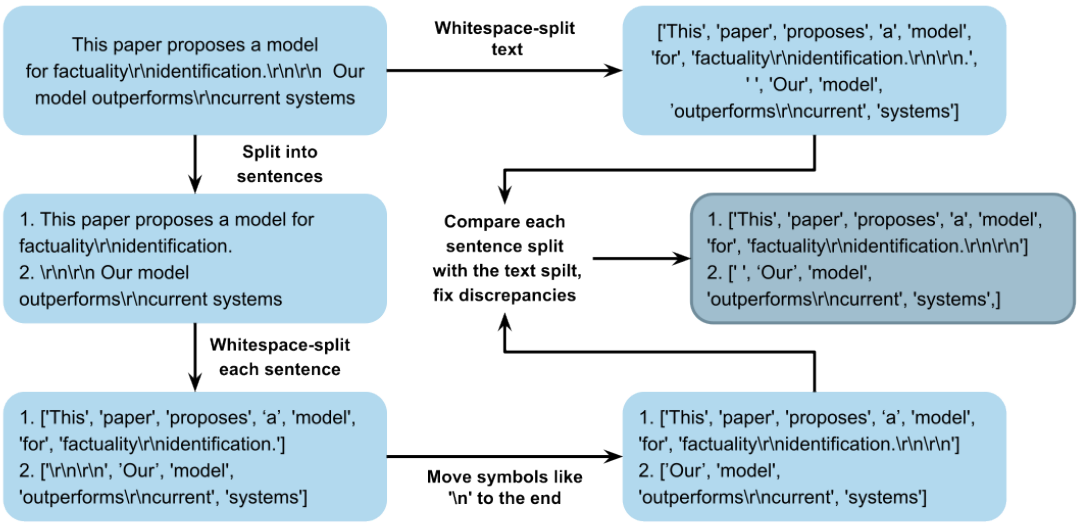


Figure 2: Preprocessing for Augmentation Pipeline

Numerical Sensitivity Enhancing and Reasoning Completeness Alignment for Quantitative Understanding

Xinyue Liang*, **Jiawei Li***, **Yizhe Yang**, **Yang Gao**†
School of Computer Science and Technology,
Beijing Institute of Technology, Beijing, China
Beijing Engineering Research Center of High Volume Language Information
Processing and Cloud Computing Applications, Beijing, China
Beijing Institute of Technology
Southeast Academy of Information Technology, Putian, Fujian, China
{xyliang, jwli, yizheyang, gyang}@bit.edu.cn

Abstract

In this paper, we describe the methods used for Quantitative Natural Language Inference (QNLI), and Quantitative Question Answering (QQA) in task1 of Semeval2024 NumEval. The challenge's focus is to enhance the model's quantitative understanding consequently improving its performance on certain tasks. We accomplish this task from two perspectives: (1) By integrating real-world numerical comparison data during the supervised fine-tuning (SFT) phase, we enhanced the model's numerical sensitivity. (2) We develop an innovative reward model scoring mechanism, leveraging reinforcement learning from human feedback (RLHF) techniques to improve the model's reasoning completeness. The experimental results demonstrate that our methods achieve outstanding performance. Our code could be found at https://github.com/Bit-numeval/NumEval.

1 Introduction

Numeral-aware language understanding plays a crucial role in narratives when contexts contain numerical comparison, implication, definition etc (Chen et al., 2023a). This importance is especially revealed in real-world applications, such as law, finance, and clinical documentation (Chen et al., 2024b). Large Language Models (LLMs) demonstrated impressive performance in various NLP tasks, but they still suffer from challenges such as hallucination of weak numerical reasoning (Zhuang et al., 2024). In this paper, we delve into solving the numerical hallucination problem, evaluated by the QNLI (Ravichander et al., 2019) and QQA (Mishra et al., 2022) subtasks in the Quantitative Understanding (QU) task (Ravichander et al., 2019) of the Semeval Numeval Task (Chen et al., 2024a). QNLI involves inferring relationships based on numerical clues, and QQA requires quantitative reasoning. Table 6 in Appendix A.1 shows examples of each task.

QQA

prompt: I will first raise a question and then provide two options. Please choose the correct answer after providing the inference process step by step, in the format of 'the answer is option 1. ####_1'.
question: "The ranger and the rustler both were riding horses that galloped at the same speed. The rustler left at 01:00 where as the ranger left at 05:00. Who has traveled further??"
option1: the ranger
option2: the rustler
Response: Let's think step by step.

label: the answer is option 2. #### 2

Solution with inaccurate steps
[1]. The ranger and the rustler both were riding horses that galloped at the same speed. The rustler left at 01:00 where as the ranger left at 05:00 hours. [2]. Since the ranger left at 05:00 hours and the rustler left at 01:00 hours, **the ranger has traveled for 4 hours more than the rustler.** [3]. As they were riding at the same speed, the ranger has traveled further in the given time period. [4]. Therefore, the answer is option 1. #### 1

Solution with incomplete reasoning
[1]. Given that both the ranger and the rustler were riding horses at the same speed. The rustler left at 01:00 and the ranger left at 05:00 hours. [2]. Therefore, the answer is option 2: the rustler. #### 2

Correct solution
[1]. Given that both the ranger and the rustler were riding horses at the same speed, the distance traveled is directly proportional to the time traveled. Distance = Speed * Time. [2] . The rustler left at 01:00 and the ranger left at 05:00 hours. Since the rustler started 4 hours earlier, the rustler has traveled for 4 hours more compared to the ranger. [3] . As they were riding at the same speed, the rustler has traveled further in the given time period. [4] . Therefore, the answer is option 2: the rustler.#### 2

Figure 1: An example from QQA dataset. LLMs may not be able to generate an accurate and complete process during quantitative reasoning. Specifically in this example, the first solution has an error in step[2] where the model confuses the concept of time period and time point, resulting in a wrong answer. And the second solution simply jump to the final answer after summarizing the problem, which is incomplete and unreasonable.

Based on our investigation and preliminary evidence of promise, we attribute LLMs' limitations on the QU tasks to two key aspects: (1) *Numerical Sensitivity*: LLMs, trained on vast quantities of text, often fail to accurately capture numerical information (Chen et al., 2023b). (2) *Reasoning Accuracy and Completeness*: as illustrated in Figure 1, LLMs may struggle to generate a comprehensive and precise step-by-step reasoning process, particularly in numerical reasoning contexts (Bílková et al., 2023).

To improve models' numerical sensitivity, Chen et al. (2023b) fine-tuned them using the Comparing

*Equal contribution.
†Corresponding author.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1830–1841

Numbers Dataset, which comprises numerical comparison statements. However, solely tuning models using the comparing number data may lead to an overfit issue. Meanwhile, recent efforts on enhancing reasoning accuracy such as process supervision by reinforcement learning on every reasoning step (Lightman et al., 2023). Nevertheless, in our cases, numerical reasoning involves a variable number of reasoning steps. Therefore, multiplying reward scores for each step (Lightman et al., 2023) reduces the overall multi-step reasoning score, which results in incomplete reasoning steps.

To address these limitations, we propose utilizing numerical comparisons of real-world contexts for more robust fine-tuning. In addition, we introduce a reasoning completeness reward designed to improve the precision of viable reasoning processes. The contributions of this paper include: (1) By integrating the comparing numbers task during the fine-tuning, we enhance the model's numerical sensitivity. Specifically, we use GPT-3.5 to integrate comparing numbers data into the real-world context, effectively preventing overfitting during training. Additionally, we reduce the long-tail effect by balancing between comparing numbers data and QU task data. Ablation studies show significant performance improvements with this method. (2) To the best of our knowledge, our study is the first time to enhance the model's reasoning completeness by RLHF. By introducing a fine-grained Reasoning Completeness Reward method, we emulate the complexity of human reasoning processes, aligning the model's accuracy and step rationality with human feedback. Experimental results confirm that our approach effectively improves the performance by ensuring a reasonable number of reasoning steps. (3) Our approach outperforms the other models of the same size across all test datasets, demonstrating strong generalizability. Furthermore, even compared to the state-of-the-art LLMs such as GPT-3.5 (Ouyang et al., 2022) and Llama2-70B (Touvron et al., 2023), our method also achieves better performance on four datasets.

2 System Overview

As shown in Figure 2, we highlight to enhance the model's *numerical sensitivity* and *reasoning completeness*. Specifically, we first use GPT-3.5 (Ouyang et al., 2022) to extend comparing numbers data into real-world contexts and fine-tune the model with this data to enhance numerical sensitivity. To prevent model overfitting, we mix 50% of the QNLI and QQA task data and 50% comparing numbers data into the SFT training dataset. Furthermore, we employ RLHF method to align every reasoning step with human-labeled process supervision. To leverage a more profitable Reward model for RLHF, we manually score the reasoning steps of the augmented positive and negative cases. In particular, we propose a newly Reasoning Completeness Reward for the PPO algorithm to encourage a complete reasoning procedure. The following subsections will detail our method.

2.1 Comparing Numbers Task for Numerical Sensitivity Enhancement

The comparing numbers task is proven to enhance the numerical sensitivity of the model (Chen et al., 2023b). Nevertheless, traditional comparing numbers data only involves the comparison of two numbers and lacks real-world contexts, which can lead to model overfitting and impairing its comprehension and generation capabilities. To address this, we use GPT-3.5 to put comparing numbers data into real-world contexts for training. Additionally, we introduce training data balance to avoid overfitting and long-tail problems. The following will provide a detailed explanation.

Comparing Numbers in Real-world Contexts The comparing numbers task was first proposed by Chen et al. (2023b), statements in the format "[Num 1] is equal to [Num 2], the answer is True/False." We further improve the statements by using GPT-3.5 to incorporate comparing numbers data into real-world contexts, thereby increasing the diversity and reality of the data. Additionally, randomly generating [Num 1] and [Num 2] overlooks the realistic numerical ranges in real-world contexts (e.g. human ages cannot reach 100,000 years old). Therefore, we restrict the numerical range to ensure that 90% of the numbers are randomly generated within the range of 0 to 10,000, thus aligning more closely with real-world contexts. Details and an example are shown in Appendix A.

Training Data Balance To balance the data and avoid long-tail problems caused by varying dataset sizes in QNLI and QQA tasks, we generate additional data by using GPT-3.5, which has increased the number of cases in each dataset to approximately 1000. Moreover, during the training phase of the comparing numbers task, we mix 50% of

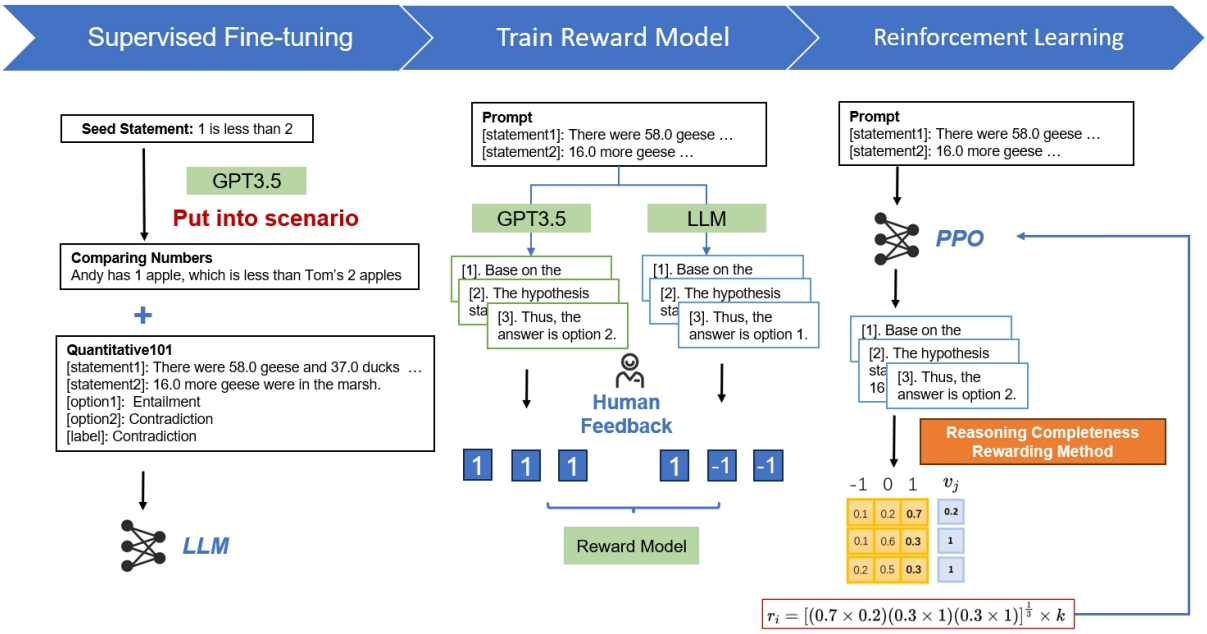


Figure 2: An overview of our system: (1) supervised fine-tuning with comparing numbers task for numerical sensitivity enhancement, (2) reward model training. (3) reinforcement learning via proximal policy optimization with Reasoning Completeness Reward.

the QU training data and 50% comparing numbers data to avoid overfitting the model to the comparing numbers task. Details on the specific expansion methods and prompt specifics can be found in the Appendix C.

2.2 RLHF-based Reasoning Completeness Alignment

To enhance the model's reasoning accuracy and completeness, we first employ human-labeled process supervision signals to align every reasoning step generated by the LLMs; then, we propose a new Reasoning Completeness Reward (RCR) model to improve the RLHF's performance to encourage generating complete reasoning steps.

2.2.1 Human-data Collection for Training Reward Model

To train a profitable reward model (RM), balanced labels need to be collected. While the number of positive labels far exceeds other labels among the steps generated by GPT-3.5, we have also used other open-source LLMs, such as Abel-7b, to generate candidates of reasoning steps, which may contain more negative examples to balance the labels' polarities. Human labelers would evaluate the given steps by their correctness, and correct answers to the question are provided as a reference. The statistics of datasets are shown in Table 1.

Datasets	Cases	Human labeled			
		Pos.	Neu.	Neg.	Steps
AwpNLI	1622	4334	822	1669	7109
NewsNLI	1643	3358	910	2870	7502
RedditNLI	1152	3074	507	958	4674
RTE_Quant	1324	3363	290	914	4817
StressTest	1369	2598	723	1921	5696
QQA	1394	3937	184	1778	6424

Table 1: The step data labeled by human. "Cases" is the number of solutions generated by models, "Pos.", "Neu.", and "Neg." are the number of positive, neutral, and negative labels after labeling, respectively, "Steps" is the total number of reasoning steps taken to solve all the questions in the dataset.

Step Labelling Criteria Each step is classified as either 'positive', 'neutral', or 'negative' due to its correctness, corresponding to three labels: "1", "0", and "-1". The correct steps must first meet the requirements of accurate logic and calculation within the steps (correct object of operation and correct result). At the same time, it is necessary to be consistent with and correctly use the results of the previous step for subsequent reasoning. If the correct conditions are met but there is no help in obtaining the correct answer, 0 points will be given. On this basis, if the task requirements are correctly understood and helpful in obtaining the correct answer, 1 point can be given. Steps with

logical, computational, or factual errors that are completely unrelated to the context and question, or incorrect answers, will receive a score of -1.

2.2.2 Reasoning Completeness Reward

Lightman et al. (2023) proposed a process supervision method by scoring the correctness probability of each reasoning step. The score is implemented as the multiplication of probabilities of all reasoning steps:

$$r_i^1 = \prod_{j=1}^{N} P(y_j = 1|x_j) \tag{1}$$

where N is the number of steps for the i-th solution, x_j is the input of RM, and y_j is the classification.

However, when the number of reasoning steps is not fixed, the score of (1) is influenced by the number of reasoning steps. As the correctness probability is decimal, the more steps involved in reasoning, the smaller the product of probabilities, resulting in lower rewards, which leads to a tendency for the model to subsequently generate less reasoning steps. To mitigate this, we applied geometric mean to the product:

$$r_i^2 = \left(\prod_{j=1}^{N} S_j \right)^{\frac{1}{N}} \tag{2}$$

where $S_j = P(y_j = 1|x_j)$ is the score for j-th step.

We observed that despite using the scoring method of (2), the model still failed to generate complete reasoning steps. Further analysis revealed that the model often simply repeats the question in its first reasoning step, resulting in a high score for the first step, which in turn leads the model to refrain from generating subsequent steps. Therefore, we propose the **reasoning completeness reward**, including a weighted geometric mean and a penalty coefficient. First, the importance of steps at different positions can be adjusted by setting weight v_j.

$$r_i^3 = \left(\prod_{j=1}^{N} v_j S_j \right)^{\frac{1}{N}} \tag{3}$$

In addition, as we hope that the solutions are around 4 steps, and solutions guessing the result from the first step without reasoning is not encouraged, a penalty coefficient k is introduced to constrain it.

$$k = \begin{cases} \frac{5}{\sigma\sqrt{2\pi}} e^{-\frac{(N-\mu)^2}{2\sigma^2}} & , N > 1 \\ 0 & , N \leq 1 \end{cases} \tag{4}$$

where $\mu = 4$, $\sigma = 2$. So the reward from the reward model is

$$R_i = r_i - \beta KL(x, y) \tag{5}$$

$$r_i^4 = \left(\prod_{j=1}^{N} v_j S_j \right)^{\frac{1}{N}} \times k \tag{6}$$

where $KL(x, y)$ is the KL-divergence between the current policy and the reference model in reinforcement learning.

Upon achieving the RM, we employ RLHF with PPO (Schulman et al., 2017) in a step-by-step manner, which is implemented with TRL[1].

3 Experimental Setups

Datasets We adopted the Quantitative101 dataset provided for SemEval 2024 Task7 and then expanded it using the GPT-3.5 API, in Table 1. From these data, three datasets were obtained for SFT, reward model training, and reinforcement learning, respectively. The prompts used during training and testing can be found in the Appendix D. Due to a large amount of labeled "1" data in the RM training dataset, each step of "0" and "-1" was repeated 2-3 times, resulting in 16587 positive steps, 11072 neutral steps, and 16236 negative steps in total. When dividing the datasets, 20% of the data is used as test sets in all three periods.

Metrics and Parameters setting The metric is the average micro-F1 score of the testing dataset in QNLI and QQA tasks. Our CN-SFT model is trained on Abel-7B (Chern et al., 2023) with a learning rate of 3e-5, a warmup rate of 0.03, and a model max length of 1024. As for the RM, we choose to train on BERT-large model (Devlin et al., 2018) as it well complete the classification tasks (Gao et al., 2022). It is trained with a learning rate of 2e-5, warmup rate of 0.05, and a model max length of 256, and is trained for 10 epochs. The PPO training is implemented with Lora, where the learning rate=1.41e-5, max new tokens=512. On a dataset of size 5470, each training epoch takes around 55 hours on 4 A100s.

4 Experimental Results

4.1 Overall Results

Main Results Table 2 compares the performance of our method with that of current mainstream

[1] https://huggingface.co/docs/trl/main/en/index

Models	QNLI					QA	Score
	AwpNLI	NewsNLI	RedditNLI	RTE_Quant	StressTest		
Llama-7B	1.47%	0.47%	0.40%	0.86%	1.36%	3.70%	1.38
GPT-3.5	42.07%	58.55%	32.0%	55.88%	33.1 %	40.12%	43.62
BLOOMZ	48.04%	54.46%	37.2%	47.64%	31.22%	51.85%	45.07
Abel-7B	55.82%	50.75%	47.20%	56.67%	30.87%	48.14%	48.24
ChatGLM	72.55%	70.42%	55.2%	60.94%	37.15%	53.70%	58.33
GPT-3.5*	77.93%	51.3%	59.2%	73.53%	**54.77%**	**63.58%**	63.39
Llama-70B	77.45%	69.01%	67.2%	73.39%	37.15%	59.26%	63.91
CN-SFT-7B	71.08%	66.67%	64.40%	72.53%	52.74%	56.17%	63.93
CN-PPO-7B	**87.25%**	**71.36%**	**75.20%**	**86.99%**	53.57%	56.68%	**71.84**

Table 2: Performance of baseline models. The prompt of GPT-3.5* has added explanations for options such as "entailment" compared to GPT-3.5. The CN means comparing numbers. CN-PPO-7B is trained on CN-SFT-7B with RCR-improved RLHF.

Dataset	Lightman et al. (2023)	Ours
AwpNLI	83.33.%	**87.25%**
NewsNLI	69.95%	**71.36%**
RedditNLI	63.20%	**75.20%**
RTE_Quant	**88.41%**	86.99%
StressTest	37.32%	**53.57%**
QQA	51.23%	**56.68%**
Score	65.57	**71.84**
Steps (avg)	2.624	**2.844**

Table 3: Comparison results indicate that our proposed RCR-improved RLHF outperforms over all datasets and can generate more completed reasoning steps.

LLMs on the QU tasks. Our model achieved optimal performance in the AwpNLI, NewsNLI, RedditNLI, and RTE_Quant tasks. It also showed comparable results in the StressTest and QA tasks, only falling short of Llama2-70B and GPT-3.5. However, it is worth emphasizing that our model has only 7B parameters. At this scale, its performance significantly surpasses that of other models.

Specifically, compared to our baseline model Abel-7B, by solely employing the CN-SFT method, our model achieved significant accuracy improvements of 15.26%, 15.92%, 17.12%, 15.86%, 21.87%, and 8.03% across six tasks. Upon further integrating the RLHF, the accuracy additionally gained 16.17%, 4.96%, 10.8%, 14.46%, 0.83%, and 0.51% improvement. These results validate the effectiveness of the methods proposed in this study.

The Effect of the Reasoning Completeness Reward (RCR) It is aimed at enhancing the completeness of the reasoning steps. Table 3 shows the comparison of our method's effectiveness and the number of reasoning steps against the baseline. The results demonstrate that the proposed RCR significantly increases the performance. Furthermore, the number of reasoning steps generated by our proposed enhances the reasoning completeness indicated by reasoning steps.

4.2 Ablation Analysis

We further conduct ablations to analyze the contribution of our methods' components.

Comparing numbers task can enhance the model's numerical sensitivity. We first verify whether the comparing numbers task enhances the model's numerical sensitivity. As shown in Table 4, By comparing the results of SFT (column 2) and CN-SFT (column 3) as well as PPO (column 4) and CN-PPO (column 5), we observe that models integrating the comparing numbers task exhibit superior performance in all datasets.

RLHF-based reasoning completeness alignment is valid. As shown in Table 4, the comparison of the PPO (column 4) to the SFT (column 2), and the comparison of the CN-PPO (column 5) to the CN-SFT (column 3) indicate that reasoning completeness alignment based on the proposed RLHF can effectively improve the model's performance on numerical understanding.

4.3 Comprehensive Analysis

4.3.1 Error Analysis

As shown in Table 2, the system performs relatively weakly on the QQA and StressTest datasets. The weak accuracy in the QQA task may be attributed to a lack of physical common sense in our 7B LLM. For instance, the question `"An apple is`

Dataset	SFT (w/o CN and RL)	CN-SFT (w/o RL)	PPO (w/o CN)	CN-PPO
AwpNLI	58.82%	**71.08%**	80.67%	**87.25%**
NewsNLI	55.87%	**66.67%**	59.62%	**71.36%**
RedditNLI	51.60%	**64.40%**	71.20%	**75.20%**
RTE_Quant	68.40%	**72.53%**	80.72%	**86.99%**
StressTest	52.57%	**52.74%**	53.40%	**53.57%**
QQA	50.62%	**56.17%**	**59.26%**	56.68%
Score	56.31	**63.93**	67.48	**71.84**

Table 4: Ablation studies of our method. SFT means the model is fine-tuned only on QU training data, while PPO refers to reinforcement learning training based on this model. CN-SFT means the model was fine-tuned on both QU training data and comparing numbers data, and CN-PPO refers to reinforcement learning training based on this model.

`sitting 15 meters away from Harry, and a watermelon is sitting 110 cm away. Which item looks larger?`". Another example is shown in Appendix B.1. Solving such a problem not only relies on numerical logical reasoning, but also requires understanding the conversion relationship between 'meters' and 'cm', and the physical principle that objects appear smaller the further away they are. This common sense is often acquired by knowledge injection for LLMs, which is out of our research scope in this paper.

The objective of the StreesTest dataset is to determine the relations of two sentences. Most of the sentences always contain multiple numbers whereas only one or two numerical information is valuable for classifying the sentences' relations. An example is shown in Appendix B.2. However, our models as well as other LLMs (i.e. GPT3.5 and Llama-70B) hardly capture the most valid numbers to predict the outcomes. As a result, the improvement of our model on the StressTest dataset is not as significant as in other datasets.

4.3.2 Strengths and Weaknesses

Strengths This study firstly integrates comparing numbers data into real-world contexts, thereby avoiding model overfitting and the deterioration of linguistic capabilities typically caused by solely using formatted data. This approach not only enhances the model's numerical sensitivity but also effectively prevents overfitting issues. Moreover, we propose a new reasoning completeness reward scoring method, suitable for more complex reasoning tasks, particularly those featuring a variable number of reasoning steps. The effectiveness of this method lies in rewarding each step of reasoning and considering the number of reasoning steps into the reward calculation, thus preventing the generation of reasoning processes that are either too brief or excessively lengthy. Finally, In the majority of tasks, our 7B model outperforms super LLMs such as GPT-3.5 (Ouyang et al., 2022) and Llama2-70B (Touvron et al., 2023).

Weaknesses First, in Section 4.3.1, We noted that the model generates incorrect answers for certain tasks due to the absence of essential physical common sense and demonstrates suboptimal performance in identifying and predicting relationships involving multiple numbers. Second, our approach substantially mitigates the model's hallucination of weak numerical reasoning but doesn't eliminate the hallucination that existed in LLMs' outcomes. Third, this study employs the PPO algorithm for the RLHF to validate its effectiveness. Nevertheless, the learning efficiency and convergence problems of the PPO algorithm have not been fully explored.

Therefore, future work is directed to the following aspects. The first one is knowledge injection (Lauscher et al., 2020; von Rueden et al., 2023), especially numerical-relevant knowledge, could be further employed to improve the numerical-aware language understanding capability of the LLMs. Second, the most valuable numbers during the reasoning process could be identified and weighted. Third, employing Score Normalization and Clipping to constrain the reward scores can resolve the training instability (Zheng et al., 2023). Last, utilizing the DPO algorithm (Rafailov et al., 2023), which implements an implicate reward, enhances training stability and its efficiency.

5 Conclusion

In this paper, we described the systems used for QNLI, and QQA in task1 of Semeval2024 NumEval. We select the Abel-7B model as the baseline model. To address the quantitative understanding problem, we first integrate comparing numbers data from real-world contexts to enhance the model's numerical sensitivity. During this process, we devise an effective data mixer to prevent overfitting and the long-tail problem. Subsequently, by employing process supervision from human feedback, we develop an innovative reward model scoring mechanism to improve the model's reasoning completeness using RLHF. Test results demonstrate that our 7B model exceptionally outperformed, surpassing LLMs such as GPT-3.5 on 4 tasks and Llama2-70B on 6 tasks, respectively.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No: 92370110, U21B2009). We appreciate the helpful discussions with Siming Liu. We also thank all the anonymous reviewers for their insightful suggestions.

References

Marta Bílková, Sabine Frittella, Daniil Kozhemiachenko, and Ondrej Majer. 2023. Qualitative reasoning in a two-layered framework. *International Journal of Approximate Reasoning*, 154:84–108.

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024a. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Chung-Chi Chen, Yu-Lieh Huang, and Fang Yang. 2024b. Semantics matter: An empirical study on economic policy uncertainty index. *International Review of Economics Finance*, 89:1286–1302.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023a. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 69–77. Association for Computational Linguistics.

Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi, and Yusuke Miyao. 2023b. Improving numeracy by input reframing and quantitative pre-finetuning task. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 69–77, Dubrovnik, Croatia. Association for Computational Linguistics.

Ethan Chern, Haoyang Zou, Xuefeng Li, Jiewen Hu, Kehua Feng, Junlong Li, and Pengfei Liu. 2023. Generative ai for math: Abel. `https://github.com/GAIR-NLP/abel`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Leo Gao, John Schulman, and Jacob Hilton. 2022. Scaling laws for reward model overoptimization.

Anne Lauscher, Olga Majewska, Leonardo FR Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. 2020. Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers. *arXiv preprint arXiv:2005.11787*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan,

Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Laura von Rueden, Jochen Garcke, and Christian Bauckhage. 2023. How does knowledge injection help in informed machine learning? In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023. Secrets of RLHF in large language models part I: PPO. *CoRR*, abs/2307.04964.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2024. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36.

A Construction Process for Our Comparing Numbers Task

To create our Comparing Numbers data, we first automatically generate seed statements and then put them into natural language paragraphs by GPT3.5.

As Table 5 shows, there are three templates for seed statements. We randomly select two numbers from 0 to 9,999 and insert them into the template, note that the distributions of each template and answers are balanced. Finally, 5059 instances are obtained, small amount of duplication in numbers is acceptable as they will be placed into different scenarios afterwards.

Considering most scenarios in the QU tasks are daily situation and financial news, we adopted the following two prompts to generate statements respectively.

Prompt for daily situations: Rewrite the sentence containing numerical comparison relationships into a paragraph describing daily situations about numbers, with a length of no more than 50 words, comparative relationships must be included: (seed statement). For example : 'There were 128,695 students in the large university, which exceeded the 107,736 count of another university.'

Prompt for financial news: Rewrite the sentence containing numerical comparison relationships into a paragraph of financial news, with a length of no more than 50 words, comparative relationships must be included : (seed statement) For example: 'In the stock market, stock A's price at 183.146 increase, surpassing stock B's price at 115.877.'

A.1 Examples of Different Tasks

As shown in Table 6, the comparing numbers task involves a statement with a numerical relationship, which requires the model to determine if it is true.

In the QNLI task, there are two statements, the first is the premise, and the second is the hypothesis. The model needs to determine the correct relationship (entailment/neutral/contradiction) between the two statements, that is, to determine whether the hypothesis can be inferred from the premise. In the QQA task, there is a question with two options, and the model's task is to work out the correct answer.

These tasks require models to interpret quantities expressed in language, perform basic calculations, judge their accuracy, and justify quantitative claims using both verbal and numeric reasoning.

B Examples of Model Results

B.1 Example from QQA task

As shown in Table 7, in QQA tasks, the model sometimes becomes confused about the knowledge required for this problem, unable to analyze based on common sense that lightweight paper airplanes can fly faster, but instead conducts analysis unrelated to the problem, resulting in incorrect answers.

B.2 Example from StressTest

From Table 8 we can see that although the model correctly extracted quantitative information, it misses the key numeral and is distracted by the text, conducting calculations unrelated to the question, resulting in wrong answer.

C Dataset Extending

C.1 QNLI tasks

For the QNLI task, first automatically generate a set of numerals which will be contained by the premise and generate the premise with GPT3.5, then rewrite the statement based on the "entailment", "neutral"or "contradiction"relationship as hypotheses.

For example, when expanding the NewsNLI dataset, we use the following prompts in sequence.

To generate a premise: " Write a piece of news in 30 words or less that contains the message "[number]"

To generate an entailed hypothesis: "Abbreviate this paragraph and keep its original meaning unchanged:" [premise] "

To generate a neutral hypothesis: " Add some numerical information to this paragraph:[statement] "

If a contradicted statement needs to be generated, simply replace the numbers in the premise, such as replacing" 30 people "with" 40 people "," more than 50 people ", or" less than 20 people ".

When expanding the AwpNLI dataset, we can first generate a pair of statements with entailment relationships, the prompt is as follows: "Generate two statements, the first being a promise that contains some quantitative information, and the second statement is a quantitative inference based on the premise. For example: [statement1]: A restaurant baked 5.0 cakes during lunch and sold 6.0 during dinner today and the restaurant baked 3.0 cakes yesterday." [statement2]: 2.0 cakes are left," and

Seed statement	Question	Label
200 **is less than** 215	At the cafe, a line of 215 individuals formed, exceeding the queue at the bakery, where 200 people were waiting.	True
83.146 **is larger than** 115,899	In the stock market, stock A's price at 83.146 increase, surpassing stock B's price at 115,877.	False
147,254 **is equal to** 32.567	There were 147,254 votes for candidate A, which was equal to 32,567 votes for candidate B.	False

Table 5: Examples of our Comparing Numbers task. The seed statements and labels are generated by randomly selecting two numbers between 0 and 9,999 to create comparison statements. The questions are then formulated by GPT-3.5 by specific prompts.

Task	Question	Label
Comparing Numbers	At the cafe, a line of 200 individuals formed, exceeding the queue at the bakery, where 215 people were waiting.	True/False
QNLI	statement1: The fighting ended with all seven attackers dead, Afghan officials said. statement2: All seven militants are dead , authorities say.	Entailment/ Contradiction/ Neutral
QQA	The beauty queen glided across the marble floors with a speed of 15 m/s but at a speed of 1000 cm/s on the wooden floors because it had? Option1: more resistance. Option2: less resistance	Option1

Table 6: Examples of Comparing Numbers, QNLI, and QQA task.

then replace the numerical information in statement 2 to obtain the conflicting statement.

C.2 QQA task

For QQA tasks, most of the questions in the dataset are simple physics multiple choice questions, including knowledge of friction and kinematics. To maintain a balanced proportion of answers, specify the answer option in the prompt of the generated question: "Generate a physics multiple choice question with a quantitative relationship, consisting of two options. The correct answer is option [ans], avoid using markers or latex markers. For example:" question ":" James is watching it rain one day. He notes that the rain travels over the driveway in 5 seconds and the bump "shingles of his house's roof in 10 seconds, the driveway and the shingles are all similar in size. Which surface is more smooth?," option1 ":" roof "," option2 ":" driveway. ".

D Prompts Used in Our Work

Table 9 shows all the prompts we used during the training process.

QQA
question: Two paper planes are thrown at a time, into the air. One is made of thick cardboard and has a weight of 200 gms, one is made of thin paper and weighs 40 gms. Which paper plane flies faster? **option1:** cardboard plane **option2:** paper plane **label:** Thus, the answer is option2. #### 2
SFT output [1]. The flight speed of a paper plane could be given by the equation: S = faultincline * cos(theta) * times filled, where fill time = t, S is the flight speed, and proprietor elevation. [2] . As the Fuji Schoolcogel 2424 AppColl Maybe is a smaller number, we can assume it's an inertial force. [3] . We can calculate the weight using the equation: W = m * g / W, where W is the wheel weight, m is the mass, g is the acceleration, and D is the diameter of the wheel. [4] . Finally, we can determine the acceleration of a cardboard plane using the equation: Continuous acceleration = (m * g) / times filled, where W is the weight, m is the mass, and D is the diameter of the wheel, w_c = 200 g. [5]. Calculating the acceleration of the cardboard plane: Continuous acceleration = (200 g * g) / 10 sec = 200 g * g / 10 sec = 200(g) / 10 sec. [6] . Calculating the acceleration of the thin paper plane: Continuous acceleration = (40 g * g) / 10 sec = 40 g * g / 10 sec = 40(g) / 10 sec. [7] . There is no difference in acceleration between the cardboard plane and the thin paper plane, so they will fly at the same speed. Thus, the answer is option 1. #### 1

Table 7: An error example from QQA task. Model sometimes becomes confused about the knowledge required for this problem, unable to analyze based on common sense that lightweight paper airplanes can fly faster, but instead conducts analysis unrelated to the problem, resulting in incorrect answers.

StressTest
statement1: If Danny **scored 86** , 75 , 52 , 87 , 85 and 82 marks (out of 100) in English , Social Studies , Art , Music , Biology and French **statement2:** If Danny **scored more than 76** , 75 , 52 , 87 , 85 and 82 marks (out of 100) in English , Social Studies , Art , Music , Biology and French **option1:** Entailment **option2:** contradiction **option3:** neutral **label:** Thus, the answer is option1. #### 1
CND-PPO output [1]: Given the numbers 86, 75, 52, 87, 85, and 82, it can be calculated that the average mark is (86 + 75 + 52 + 87 + 85 + 82) / 6 = 485 / 6 = 77.5.[2] : The hypothesis states that Danny scored more than 76, 75, 52, 87, 85, and 82 marks.[3] : As the average mark approximately equals 77.5, and each mark is rounded to the nearest whole number before factoring, the hypothesis is 'not car more than 76, 75, 52, 87, 85, and 82 marks' in the given context.Thus, the answer is option 3. #### 3 **reward:** 0.28299634026503834
Correct solution [1]: The premise states that Danny scored 86 , 75 , 52 , 87 , 85 and 82 marks (out of 100). The hypothesis states that Danny scored more than 76, 75, 52, 87, 85, and 82 marks. [2]: **86 is indeed more than 76**, so the hypothesis can be infered to be true. [3] Thus, the answer is option 1. #### 1

Table 8: An error example from StressTest dataset. Although the model correctly extracted all the quantitative information, it misses the key point and conducted analysis and calculations unrelated to the question, resulted in wrong answer.

QQA I will first raise a question and then provide two options. Please choose the correct answer after providing the inference process step by step, in the format of 'the answer is option 1. #### 1'. If calculation is involved, please provide the equations during the calculation process. Using numbers like '1.' or '[1]' to mark steps. question: option1: option2: Response: Let's think step by step.
AwpNLI I will first raise two statements and then provide two options which are entailment and contradiction. The first statement is the given premise, the second statement is the hypothesis. You should determine if the hypothesis can be justifiably inferred to be true (option 1 : entailment) or false (option 2 : contradiction) base on the premise. Please choose the correct answer after providing the inference process step by step, in the format of 'the answer is option 1. #### 1'. If calculation is involved, please provide the equations during the calculation process. Using numbers like '1.' or '[1]' to mark steps. Choose the correct answer in the format of 'the answer is option 1. #### 1'. statement1: statement2: option1: option2: Response: Let's think step by step.
NewsNLI I will first raise two statements and then provide two options which are entailment and neutral. The first statement is the given premise, the second statement is the hypothesis. You should determine if the hypothesis can be justifiably inferred to be true (option 1: entailment) or cannot be determined (option 2: neutral) base on the premise. You should pay attention to additional information rather than shared information, especially paying attention to whether the numbers are reasonable and derived from the premise. If there is information that is not mentioned in the premise or cannot be directly inferred from the hypothesis, then the hypothesis cannot be determined. Please choose the correct answer after providing the inference process step by step, in the format of 'the answer is option 1 #### 1'. Using numbers like '1.' or '[1]' to mark steps. statement1: statement2: option1: option2: Response: Let's think step by step.
RTE I will first raise two statements and then provide two options which are entailment and neutral. The first statement is the given premise, the second statement is the hypothesis. You should determine if the hypothesis can be justifiably inferred to be true (option 1 : entailment) or cannot be determined (option 2 : neutral) base on the premise. You should pay attention to additional information rather than shared information, especially paying attention to whether the numbers are reasonable and derived from the premise. If there is information that is not mentioned in the premise or cannot be directly inferred in the hypothesis, then the hypothesis cannot be determined. Please choose the correct answer after providing the inference process step by step, in the format of 'the answer is option 1. #### 1'. Using numbers like '1.' or '[1]' to mark steps. statement1: statement2: option1: option2: Response: Let's think step by step.
RedditNLI I will first raise two statements and then provide three options which are entailment, contradiction and neutral. The first statement is the given premise, the second statement is the hypothesis. You should determine if the hypothesis can be justifiably inferred to be true (option 1 : entailment), false (option 2 : contradiction) or cannot be determined (option 3 : neutral) base on the premise. Please choose the correct answer after providing the inference process step by step, in the format of 'the answer is option 1. #### 1'. Using numbers like '1.' or '[1]' to mark steps. statement1: statement2: option1: option2: option3: Response: Let's think step by step.
StressTest I will first raise two statements and then provide three options which are entailment, contradiction and neutral. The first statement is the given premise, the second statement is the hypothesis. You should determine if the hypothesis can be justifiably inferred to be true (option 1 : entailment), false (option 2 : contradiction) or cannot be determined (option 3 : neutral) base on the premise. You should especially pay attention to whether the numbers are reasonable and derived from the premise. If there is information that is cannot be directly inferred in the hypothesis, then the hypothesis cannot be determined. Please choose the correct answer after providing the inference process step by step, in the format of 'the answer is option 1. #### 1'. Using numbers like '1.' or '[1]' to mark steps. statement1: statement2: option1: option2: option3: Response: Let's think step by step.

Table 9: Our prompts used for different datasets in the training process.

MaiNLP at SemEval-2024 Task 1: Analyzing Source Language Selection in Cross-Lingual Textual Relatedness

Shijia Zhou[1,*] **Huangyan Shan**[1,*] **Barbara Plank**[1,2] **Robert Litschko**[1,2]
[1]MaiNLP, Center for Information and Language Processing, LMU Munich, Germany
[2]Munich Center for Machine Learning (MCML), Munich, Germany
{zhou.shijia, Shan.Huangyan}@campus.lmu.de {bplank, rlitschk}@cis.lmu.de

Abstract

This paper presents our system developed for the SemEval-2024 Task 1: Semantic Textual Relatedness (STR), on Track C: Cross-lingual. The task aims to detect semantic relatedness of two sentences in a given target language without access to direct supervision (i.e. zero-shot cross-lingual transfer). To this end, we focus on different source language selection strategies on two different pre-trained languages models: XLM-R and FURINA. We experiment with 1) single-source transfer and select source languages based on typological similarity, 2) augmenting English training data with the two nearest-neighbor source languages, and 3) multi-source transfer where we compare selecting on all training languages against languages from the same family. We further study machine translation-based data augmentation and the impact of script differences. Our submission achieved the first place in the C8 (Kinyarwanda) test set.

1 Introduction

The task of semantic textual relatedness (STR) has a long-standing tradition in NLP (e.g., Mohammad, 2008). It consists of predicting a score that reflects the closeness in semantic meaning between two given sentences. For example, consider the following examples extracted from the actual shared task data (Abdalla et al., 2023) shown in Figure 1. For English, the annotators scored the first pair higher than the second sentence pair. Similarly, for Afrikaans the annotators scored the first example higher than the second one. As further described in Abdalla et al. (2023), all sentence pairs were annotated manually in a pairwise fashion to obtain semantic textual relatedness (STR) scores between 0 (completely unrelated) and 1 (maximally related).

While previous work has largely focused on English, the SemEval-2024 shared task 1 (Ousidhoum et al., 2024b) aims to extend the language coverage. It proposes datasets to evaluate the relatedness of sentence pairs for a total of 14 languages, including low-resource tail languages such as Kinyarwanda (kin) or Marathi (mar) (Abdalla et al., 2023) (see §2.1). The shared task includes three subtracks, each with a focus on supervised, unsupervised and cross-lingual STR, respectively. In this paper, we focus on Track C, *cross-lingual STR*. In this track, the goal is to develop a system to predict STR scores *without* access to any labeled data for the target language (importantly, also no target development data). That is, Track C requires the development of a regression model for 12 target languages, without relying on any labeled datasets in the target language (or pre-trained language model fine-tuned on other STR tasks). Instead the cross-lingual task allows to utilize training datasets from at least one other language from the other tracks (which includes training data of up to 9 languages). Returning to our running example in Figure 1, the task is to develop a system for example for Afrikaans as target by transferring knowledge from one or more source languages (which may include English).

Pair	STR	Sentence Pair
eng-25	0.88	"It is better known as a walk." "It is also known as a walk ."
eng-31	0.30	"But, of course, it's not that simple" "However, this is not for me."
afr-87	0.72	"ols totdat dit n bal vorm." "Dit moet n stywe bal deeg vorm."
afr-78	0.09	"Stel jou voor jou kind skryf elke week n opstel." "Washington is ook n fietsryer-vriendelike stad."

Figure 1: Examples from the dev sets for Semantic Textual Relatedness (STR). eng: English, afr: Afrikaans.

Previous work on multilingual NLP has illustrated the *curse of multilinguality* (Conneau et al., 2020), that is, diminishing returns for training a

[*] Both authors contributed equally.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1842–1853

single system on many languages due to language interference. This shared task has a focus on low-resource languages and languages typologically distant to English, a setup in where cross-lingual transfer has shown to be particularly challenging (Lauscher et al., 2020). Motivated by these two aspects, we set out to study the use of fewer but more relevant *source* languages for a given target language. More specifically, we aim to find good "donor language(s)" (Malkin et al., 2022) and compare those to baselines that either only use English, or a multi-source model trained on all source languages (except the target). We aim to answer the following research questions: **RQ1** To what extent does knowledge transfer from source languages improve STR models? **RQ2** Do multilingual STR models exhibit language interference (Wang et al., 2020), i.e., performance drops when training data from heterogeneous languages are combined? **RQ3** To what extent do script differences play a role in STR ("script gap"), and can we narrow the script gap by using a foundation model specialized to align transliterated data and data written in different scripts? **RQ4** Can we further improve the transfer performance by relying on machine translation to augment existing training data?

To study RQ1, we make use of typological information available in language vectors. For RQ2, we opt for a multi-source approach, that combines the training data for all languages (except the target). To study the impact of scripts (RQ3), we make use of transliteration, and further compare a BERT-based model to FURINA (Liu et al., 2024), a recently proposed language model that aims to better align languages across scripts. Finally for RQ4, we investigate the use of machine translation (MT) for data augmentation. We apply ours methods to 12 target languages in Track C. The specific details about languages are presented in §2.1.

2 Background

2.1 STR Task Setup and Datasets

The STR task (Ousidhoum et al., 2024b) aims to measure the extent to which two linguistic elements share semantic proximity (Ousidhoum et al., 2024a). These elements may be associated through various means, such as conveying similar ideas, originating from the same historical period, complementing each other's meaning, and so forth. It offers 3 tracks to follow: supervised (Track A), unsupervised (Track B), cross-lingual (Track C). In Track C, participants must provide systems developed without relying on any labeled datasets specifically tailored for semantic similarity or relatedness in the target language. Instead, they are required to employ labeled dataset(s) from at least one other language.

The STR task involves 14 monolingual datasets for Afrikaans (`afr`), Amharic (`amh`), Modern Standard Arabic (`arb`), Algerian Arabic (`arq`), Moroccan Arabic (`ary`), English (`eng`), Spanish (`esp`), Hausa (`hau`), Hindi (`hin`), Indonesian (`ind`), Kinyarwanda (`kin`), Marathi (`mar`), Punjabi (`pan`), and Telugu (`tel`). Among these, Track A and Track C comprise 9 and 12 languages respectively (see Table 1). In the training datasets, each instance consists of a sentence pair and is assigned a golden STR score as judged by native speakers. The score ranges between 0 and 1, with higher values indicating greater relatedness between the sentence pairs. For details on the data collection, we refer the reader to the shared task overview paper (Ousidhoum et al., 2024a).

As per requirement, we designate the 9 languages in Track A as source languages and those in Track C as the 12 target languages. An overview of the resulting train/dev/test data statistics for the 14 languages is provided in Table 1.

2.2 Evaluation Metric

The evaluation metric used in this shared task is Spearman's rank correlation coefficient. It evaluates the strength and direction of the monotonic relationship between two variables with a range from -1 to 1. In the context of our task, as previously mentioned, the scoring has been adjusted to range between 0 and 1. We use the evaluation script provided by the organizers (Ousidhoum et al., 2024b).

2.3 Baselines

The organizers fine-tuned LaBSE (Feng et al., 2022) on the English training set to get baselines for all target languages except English (cf. §3.1). For English, they fine-tuned LaBSE on Spanish as a baseline. Since the test dataset for Spanish has not been made publicly available, all models aimed at Spanish evaluation are conducted solely on their respective validation datasets. In order to ensure a more equitable comparison with other findings, we reproduce the baseline LaBSE model utilizing the methodology provided by the organizers. It yields a baseline score of 0.687 on the Spanish validation

	eng	esp	afr	hin	pan	amh	arb	arq	ary	hau	ind	kin	mar	tel	total
Train	5,500	1,562	-	-	-	992	-	1,261	924	1,736	-	778	1,200	1,170	15,123
Dev	249	139	375	288	242	95	32	97	70	212	144	222	293	130	2,588
Test	2,600	140	375	968	634	171	595	583	425	594	360	222	-	-	7,667

Table 1: STR Dataset statistics. Indo-European lanuguages including esp, afr, hin, ind, pan and mar: 10,424 train instances; 1,811 dev instances; 5,357 test instances. Afro-Asiatic languages including hau, amh, arb, ary and arq: 3,921 train instances; 411 dev instances; 2,197 test instances. Out of 14 languages, 5 languages including amh, hin, arb, arq, ary are in non-latin script, all the rest of languages are in latin script.

dataset.

3 Methods

We opt for two RoBERTa-based (Liu et al., 2019) models for the regression task trained with a mean-squared error (MSE) loss. More specifically, we use the XLM-RoBERTa base model, and FURINA (Liu et al., 2024), which is a XLM-R derivative based on Glot-500 (ImaniGooghari et al., 2023), further detailed below. We adopt a multi-source approach that involves individually fine-tuning a model for each target language in Track C. This fine-tuning process utilizes the training datasets from all languages available in Track A, explicitly excluding the dataset of the test language itself. For baseline comparisons, we use XLM-RoBERTa (Conneau et al., 2020) and FURINA (Liu et al., 2024) models fine-tuned solely on English datasets.

3.1 Model Selection

XLM-RoBERTa. The multilingual masked language model XLM-RoBERTa (XLM-R) (Conneau et al., 2020) pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages has shown superior performance compared to Multilingual BERT (mBERT) (Devlin et al., 2019) across a range of cross-lingual benchmarks. In the experiment, we utilize the base version of XLM-R.[1] XLM-R has seen all SemRelEval languages except for Algerian Arabic (arq), Moroccan Arabic (ary), Kinyarwanda (kin) at pre-training time.

FURINA. FURINA (Liu et al., 2024) covers 511 low-resource languages. It was fine-tuned on Glot500-m (ImaniGooghari et al., 2023). The training data consists of 5% of Glot500-m's pretraining sentences in original script as well as their corresponding Latin transliterations. At pre-training time FURINA has been exposed to all SemRelEval languages except for Algerian Arabic (arq).

LaBSE. The organizers provide cross-lingual baselines for each target language by fine-tuning Language-agnostic BERT Sentence Embeddings (`LaBSE`) (Feng et al., 2022), which supports 109 languages. LaBSE was pre-trained using Translation language modeling (TLM) (Conneau and Lample, 2019), which included bilingual translation sentence pairs for training. The bilingual corpus is constructed from web pages using a bitext mining system, filtered by a pre-trained contrastive data-selection scoring model, and manually curated to create a high-quality collection of 6 billion translation pairs. Out of those, LaBSE has been exposed to different amounts of parallel data (`eng-xxx`) from SemRelEval languages. The largest amount of parallel text involves Spanish with over 375M sentence pairs (`eng-esp`), followed by Indonesian with over 250M sentence pairs (eng-ind), followed by Hindi and Arabic (`eng-{hin, arb}`) with over 125M language pairs. All other languages (`afr`, `pan`, `amh`, `haus`, `tel`, `kin`, `mar`) appear in the TLM training corpus with less than 125M sentence pairs.

3.2 Source Language Selection

Single-Source Transfer. In our first approach, we follow the standard single-source zero-shot cross-lingual transfer setup and fine-tune pre-trained language models on English data ($\texttt{XLM-R}_{\texttt{eng}}$, $\texttt{Furina}_{\texttt{eng}}$). This is a common evaluation approach adopted in standard natural language understanding and generation benchmarks (Liang et al., 2020; Ruder et al., 2023). However, English has been shown to not always be the best source language (Turc et al., 2021). To investigate if this also true for SemRelEval, we further experiment with selecting for each test language its closest (i.e., most similar) source language. Here, we measure language similarity according to typological features from the lang2vec library (Littell et al., 2017).

[1] https://huggingface.co/FacebookAI/xlm-roberta-base

K-nearest-neighbor languages. In this approach we augment the English training dataset with the datasets of k languages that are closest to the target language, dubbed kNN. To determine suitable source languages for each target language, we assess language similarity by calculating the cosine similarity between language vectors learned by a multilingual neural MT model provided by Malaviya et al. (2017). We specifically use the cell_state language vectors, which are computed by encoding all sentences in a given language and then computing the average hidden cell state of the encoder LSTM.[2] These vectors can be seen as language embeddings encoding latent typology features (Östling and Tiedemann, 2017; Yu et al., 2021). With our kNN-models we aim for a good balance between large amounts of training instances (English) and positive transfer from similar languages.

Multi-Source Transfer. The STR dataset contains languages from different language families. To investigate whether training a single model on a disverse set of languages leads to negative interference (Wang et al., 2020) we compare two multi-source models. In the first model, dubbed MS-All, we fine-tune XLM-R and Furina on the concatenation of all training sets from Track A (excluding the target language). Inspired by previous work on combining *multiple related* source languages (Snæbjarnarson et al., 2023; Lim et al., 2024), we further evaluate multi-source models trained on languages from the same language family (MS-Fam).

3.3 Other Approaches

Machine Translation. For the purpose of data augmentation and balance of languages, we translate selected languages into each other using NLLB (Costa-jussà et al., 2022), ensuring that each language contributes equally to the training dataset. Taking Kinyarwanda as an example, we select Hausa and Spanish as the two languages closest to it, based on dense language vector similarity as outlined above (kNN), along with English, as training dataset. We translate among these three languages mutually, thus tripling the size of the training dataset while ensuring a balanced representation of all languages.

Transliteration. Additionally, we attempt to further facilitate multilingual transfer learning by standardizing script across languages. Utilizing the tool Uroman[3] (Hermjakob et al., 2018), which was also used by FURINA (Liu et al., 2024), we transliterate the train and test datasets of languages written in non-Latin scripts, including both the original datasets and the translated datasets, into Latin script. We evaluate the models fine-tuned on Romanized training data on the Romanized test dataset. This attempt only involves non-Latin script languages (amh, arb, ary, arq, hin).

4 Experimental Setup

The detailed setting are listed in Appendix A. As baseline, we exclusively train a model on the English dataset ($\texttt{XLM-R}_{\texttt{eng}}$, $\texttt{Furina}_{\texttt{eng}}$) and assess its performance across all target languages. Subsequently, for each target language, we fine-tune a multi-source model: if the target language is not within the 9 training datasets, we train on the union of all $n = 9$ training languages. Otherwise we train a multi-source model on $n - 1 = 8$ source languages, excluding the target ($\texttt{XLM-R}_{\texttt{MS-All}}$, $\texttt{Furina}_{\texttt{MS-All}}$). Following this, we explore whether it is helpful to prune certain languages, retaining only English and the two closest to the target languages according to lang2vec (Littell et al., 2017)[4] as source languages ($\texttt{XLM-R}_{\texttt{L2V}}$, $\texttt{Furina}_{\texttt{L2V}}$). Due to the reduction in the training set, which significantly decreased the size of the data, we attempted to expand the dataset through cross-translation ($\texttt{XLM-R}_{\texttt{L2V-Aug}}$, $\texttt{Furina}_{\texttt{L2V-Aug}}$; cf. §3.2).

5 Results and Discussion

Our main results are presented in Table 2 and are discussed in the following section.

Single-source versus multi-source transfer. We first compare the performance of a zero-shot STR model trained on English ($\texttt{XLM-R}_{\texttt{eng}}$, $\texttt{Furina}_{\texttt{eng}}$) against a multi-source model trained on the concatenation of all available languages from Track A ($\texttt{XLM-R}_{\texttt{MS-All}}$, $\texttt{Furina}_{\texttt{MS-All}}$). Our results reveal that knowledge transfer from multiple source languages (**RQ1**) improves STR models, affirming the potential of multi-source training to enhance cross-lingual capabilities. On average, both MS-All models outperform their single-source counterparts by 0.02 and 0.09 respectively. This is expected since

[2] https://github.com/chaitanyamalaviya/lang-reps/

[3] https://github.com/isi-nlp/uroman

[4] We compare the similarity of languages based on three criteria: lang_cell_states, lang_vecs and language typological vectors

	Indo-European					Afro-Asiatic					Other		
	eng	esp	afr	hin	pan	amh	arb	arq	ary	hau	ind	kin	avg
LaBSE (baseline)	0.80	0.69	0.79	0.76	-0.05	**0.84**	**0.61**	0.46	0.40	0.62	**0.47**	0.57	0.67
$\text{Furina}_{\text{eng+esp+hau}}$	-	-	0.74	0.70	**0.09**	0.73	0.40	0.27	0.57	-	0.32	0.68	-
Models based on XLM-R (Conneau et al., 2020)													
$\text{XLM-R}_{\text{eng}}$	-	0.67	**0.81**	0.80	-0.02	0.81	0.60	0.50	0.60	0.64	0.42	0.46	0.71
$\text{XLM-R}_{\text{MS-All}}$	**0.84**	0.63	0.80	**0.82**	-0.01	0.80	0.56	0.59	0.82	0.66	0.42	0.69	**0.73**
$\text{XLM-R}_{\text{MS-Fam}}$	0.82	0.71	**0.81**	**0.82**	0.00	0.69	0.44	0.37	**0.83**	0.66	-	-	0.68
$\text{XLM-R}_{\text{kNN}}$	-	0.59	**0.81**	0.78	-	0.75	0.57	-	0.50	0.62	0.45	0.41	0.69
$\text{XLM-R}_{\text{kNN+MT}}$	-	0.64	0.80	0.78	-	0.77	0.54	-	0.55	0.62	0.36	0.55	0.70
$\text{XLM-R}_{\text{kNN+TL}}$	-	-	-	0.66	-	0.37	0.45	-	0.52	-	-	-	-
Models based on Furina (Liu et al., 2024)													
$\text{Furina}_{\text{eng}}$	-	0.54	0.79	0.70	-0.14	0.74	0.37	0.45	0.59	0.63	0.44	0.53	0.62
$\text{Furina}_{\text{MS-All}}$	0.83	0.59	0.79	0.76	-0.02	0.81	0.49	**0.61**	**0.83**	0.65	0.35	**0.78**	0.71
$\text{Furina}_{\text{MS-Fam}}$	0.83	**0.72**	0.79	0.77	0.02	0.66	0.42	0.55	0.82	**0.68**	-	-	0.71
$\text{Furina}_{\text{kNN}}$	-	0.59	0.80	0.72	-	0.74	0.43	-	0.57	0.63	0.46	0.68	0.67
$\text{Furina}_{\text{kNN+MT}}$	-	0.56	0.78	0.75	-	0.74	0.44	-	0.57	0.59	0.37	0.64	0.67
$\text{Furina}_{\text{kNN+TL}}$	-	-	-	0.67	-	0.72	0.44	-	0.56	-	-	-	-

Table 2: Spearman's rank correlation of zero-shot transfer experiments on SemRelEval 9 test languages. The organizers decided to keep the test set for Spanish private, we therefore report the performance on the validation set. We exclude English from the average result (avg). **bold**: Best result for each language. Languages not covered by all L2V features are excluded from the average (eng, pan, arq, ind, kin). For our kNN-variants we opt for $k = 2$.

the multi-source training dataset is with 15,123 instances almost three times larger than the English dataset with 5,500 instances (cf. Table 1). When trained solely on English data, FURINA performs substantially worse than XLM-R. However, this performance gap narrows when transitioning from single-source to multi-source training.

Transfer from language families. After showing that models trained on all languages outperform the single-source baseline, we now investigate the effect of training on languages from the same family as source languages. Here we experiment with two multi-source models specialized only on Indo-European and Afro-Asiatic languages respectively (MS-Fam). Importantly, for each target language we train a multi-source model on all other languages in the same language family.[5] On Indo-European languages, we find that $\text{XLM-R}_{\text{MS-Fam}}$ and $\text{Furina}_{\text{MS-Fam}}$ yield similar results with much less training data (i.e., 4,913 fewer instances belonging to other language families). For Spanish, our models show performance gains of +0.8 and +0.13 for XLM-R and FURINA respectively, when compared to models trained on all languages. This underscores the presence of language interference (Wang et al., 2020) in multilingual STR models when the training data from dissimilar languages are combined (**RQ2**). On Afro-Asiatic languages, we observe average performance drops of -0.09 and -0.06 for XLM-R and FURINA when moving from MS-All to MS-Fam. We hypothesize that this can be attributed to the amount of training data available. In fact, there are 28% fewer training instances for all Afro-Asiatic languages than for English (5,500).

Transfer from nearest language neighbors. We now investigate the transfer performance when training STR models on their two closest languages according to cosine similarity of language cell state vectors, i.e. learned language vectors presented in (Malaviya et al., 2017). As mentioned earlier, we add English due to its large scale as a third training language. Our submitted system, $\text{Furina}_{\text{eng+esp+hau}}$, is trained on the two closest training languages of Kinyarwanda (kin) and has been ranked first place on the shared task leaderboard. Applying the same approach for each test language ($\text{XLM-R}_{\text{kNN}}$, $\text{Furina}_{\text{kNN}}$) shows mixed results. This indicates that the strong performance on kin can be attributed to the fact that, contrary to XLM-R, kin has been seen by Furina during pretraining.

Transliteration and cross-translation. The STR dataset contains six test languages in non-Latin scripts: Hindi (hin), Punjabi (pan), Amharic

[5] Indonesian and Kinyarwanda are the only SemRel languages in their family, we therefore cannot evaluate multi-source for those languages.

	Indo-European					Afro-Asiatic					Other		
XLM-R	eng	esp	afr	hin	pan	amh	arb	arq	ary	hau	ind	kin	avg
MIN	0.78	0.57	0.74	0.71	-0.14	0.73	0.47	0.39	0.40	0.40	0.31	0.41	0.58
XLM-R$_{eng}$	-	0.67	0.81	0.80	-0.02	0.81	0.60	0.50	0.60	0.64	0.42	0.46	0.71
XLM-R$_{kNN}$	-	0.68	0.74	0.72	-	0.75	0.57	-	0.40	0.63	0.49	0.43	0.64
L2V-Pho	0.78	0.67	0.80	0.80	-0.03	0.78	0.51	0.58	0.55	0.58	0.39	0.47	0.67
L2V-Syn	0.78	0.67	0.83	0.80	-0.03	0.74	0.51	0.58	0.55	0.61	0.39	0.43	0.67
L2V-Inv	0.82	0.63	0.80	0.80	-0.03	0.79	0.51	0.58	0.55	0.58	0.33	0.45	0.67
L2V-Fam	0.82	0.67	0.83	0.80	-0.03	0.79	0.51	0.58	0.55	0.595	-	-	0.68
L2V-Geo	0.78	0.57	0.80	0.80	-0.03	0.75	0.47	0.56	0.55	0.63	0.31	0.45	0.66
L2V-LRN	-	0.68	0.74	0.72	-	0.79	0.57	-	0.54	0.40	0.49	0.43	0.63
MAX	0.82	0.69	0.83	0.80	0.04	0.79	0.61	0.63	0.74	0.66	0.49	0.65	0.73

Table 3: Single-source transfer results in terms of spearman correlation. The language selection is based on the cosine similarity of different typological features obtained from lang2vec (L2V). We additionally report the lower (MIN) and upper bound (MAX) obtained from selecting the best and worst source language. Languages not covered by all L2V features are excluded from the average: eng, pan, arq, ind, kin. For L2V-Phon, both tel and mar are closest to hin. For L2V-Fam amh and arq are the closest languages, we report their average score (0.595). In single-source transfer with XLM-R$_{kNN}$ we use $k = 1$ and do not combine the selected language with eng training data.

(amh), Standard Arabic (arb), Algerian Arabic (arq), and Moroccan Arabic (ary). Zero-shot cross-lingual transfer of models fine-tuned on English performs worse for Arabic scripts than for amh and hin. Punjabi shows the lowest results by a large margin. When fine-tuned on on multiple source languages (MS-All), XLM-R improves the performance on four out of six languages while Furina yields improvements on all five languages. We find that (1) there is no clear winner between XLM-R and FURINA when applied on text written in different scripts, and (2) romanizing all training and test languages did not improve zero-shot cross-lingual transfer for STR (**RQ3**).

Next, we investigate the impact of augmenting the training data with translated data. The varied outcomes of augmenting data indicate that while machine translation can enhance transfer performance for certain languages. Performance drops in others may stem from shifts in label semantics and the degree of relatedness between original and translated sentence pairs (**RQ4**). Appendix C (Table 14) shows an example where MT fails to capture nuanced differences between closely, but not perfectly related sentences, leading to near-identical translations and inconsistent labels.

Single-source transfer results. We now select the most similar source languages based on different typological features obtained from the lang2vec (L2V) library. We obtain L2V vectors for Phonology (Pho), Syntax (Syn), Inventory (Inv), Family (Fam), Geography (Geo) and learned (LRN) features.

Table 3 shows our results for XLM-R.[6] Overall, a careful selection of a single-source language is crucial for zero-shot cross-lingual transfer. There is a substantial gap between the worst possible result (0.58) and the best possible result (0.73). On average, English is the most effective source language with a correlation of 0.71. A closer analysis reveals that English is the best language only for half of the target languages, despite being the language with the largest training dataset (cf. Table 13 in Appendix). Interestingly, the best possible single-source language selection (MAX) results into the same performance as XLM-R$_{MS-All}$ (cf. Table 2).

6 Conclusion

In this paper, we investigate source language selection for cross-lingual transfer for Semantic Textual Relatedness (STR). We evaluate three different language selection strategies: single-source, multi-source transfer and transfer from English and two nearest language neighbors. We find that the transfer performance crucially depends on the size of the training dataset and the linguistic proximity to the test language. We further show that script differences cause high variance transfer performance and MT-based data augmentation can lead to shifts in label semantics. Fine-tuning FURINA on eng, esp, and hau, we achieve first place in the SemEval-2024 Task 1, Track C8 (kin).

[6] FURINA results can be found in Appendix Table 12.

Acknowledgements

This research is supported by the ERC Consolidator Grant DIALECT 101043235.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proceedings of NeurIPS*, volume 32, pages 7059–7069. Curran Associates, Inc.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Ulf Hermjakob, Jonathan May, and Kevin Knight. 2018. Out-of-the-box universal Romanization tool uroman. In *Proceedings of ACL 2018, System Demonstrations*, pages 13–18, Melbourne, Australia. Association for Computational Linguistics.

Ayyoob ImaniGooghari, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Seong Hoon Lim, Taejun Yun, Jinhyeon Kim, Jihun Choi, and Taeuk Kim. 2024. Analysis of multi-source language training in cross-lingual transfer. *arXiv preprint arXiv:2402.13562*.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Yihong Liu, Chunlan Ma, Haotian Ye, and Hinrich Schütze. 2024. Translico: A contrastive learning framework to address the script barrier in multilingual pretrained language models.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535, Copenhagen, Denmark. Association for Computational Linguistics.

Dan Malkin, Tomasz Limisiewicz, and Gabriel Stanovsky. 2022. A balanced data approach for evaluating cross-lingual transfer: Mapping the linguistic blood bank. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4903–4915, Seattle, United States. Association for Computational Linguistics.

Saif Mohammad. 2008. *Measuring semantic distance using distributional profiles of concepts*. University of Toronto.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 644–649, Valencia, Spain. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.

Sebastian Ruder, Jonathan Clark, Alexander Gutkin, Mihir Kale, Min Ma, Massimo Nicosia, Shruti Rijhwani, Parker Riley, Jean-Michel Sarr, Xinyi Wang, John Wieting, Nitish Gupta, Anna Katanova, Christo Kirov, Dana Dickinson, Brian Roark, Bidisha Samanta, Connie Tao, David Adelani, Vera Axelrod, Isaac Caswell, Colin Cherry, Dan Garrette, Reeve Ingle, Melvin Johnson, Dmitry Panteleev, and Partha Talukdar. 2023. XTREME-UP: A user-centric scarce-data benchmark for under-represented languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1856–1884, Singapore. Association for Computational Linguistics.

Vésteinn Snæbjarnarson, Annika Simonsen, Goran Glavaš, and Ivan Vulić. 2023. Transfer to a low-resource language via close relatives: The case study on Faroese. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 728–737, Tórshavn, Faroe Islands. University of Tartu Library.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.16171*.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

Dian Yu, Taiqi He, and Kenji Sagae. 2021. Language embeddings for typology and cross-lingual transfer learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7210–7225, Online. Association for Computational Linguistics.

A Hyperparameters

We employed identical hyperparameters across all variants of XLM-R and FURINA. We train our models for at most 30 epochs with a batch size of 32 and a learning rate of 2e-5 and use AdamW (Loshchilov and Hutter, 2017) with a weight decay of 1e-3. We evaluate the dev set performance every 200 steps and stop early based on the spearman correlation on the validation set (patience counter: 8, patience threshold: 1e-4).

B Language Similarities

Table 4 shows for each test language its two closest source languages (`kNN`) according to cell state vectors from (Malaviya et al., 2017) and learned vectors from lang2vec (`L2V-LRN`) (Littell et al., 2017). We find both language vectors lead to similar results. Here, we further show the selected languages for our multi-source model (`MS-Fam`), which outperforms both `L2V-LRN` and `kNN`.

In Table 5-11 we show cosine similarities between all train and test languages according to different typological features extracted from L2V and learned vectors from (Malaviya et al., 2017). We use the similarities to select source languages for our `kNN` and single-source model variants.

Model Variant	Source languages	Target language	# Train Instances	FURINA	XLM-R
Based on cell state vectors (kNN) (Malaviya et al., 2017)					
1	esp, kin	afr hau	7840	0.80 0.63	0.81 0.62
2	esp, hau	ind kin	8798	0.46 0.68	0.45 0.41
3	kin, hau	amh esp	8014	0.74 0.59	0.75 0.59
4	amh, hau	ary	8228	0.57	0.50
5	kin, amh	arb	7270	0.44	0.57
6	amh, esp	hin	8054	0.72	0.78
avg	-	-	-	0.58	0.58
Based on learned lang2vec vectors (L2V-LRN) (Littell et al., 2017)					
1	esp, kin	afr arb hau ind	7840	0.80 0.46 0.63 0.44	0.81 0.60 0.62 0.39
2	esp, hau	kin	8798	0.68	0.41
3	kin, hau	amh esp	8014	0.74 0.59	0.75 0.59
4	amh, hau	hin	8228	0.74	0.79
5	kin, amh	ary	7270	0.52	0.55
avg	-	-	-	0.59	0.60
Based on language familis features (MS-Fam)					
1	esp, mar, tel	eng	3932	0.83	0.82
2	eng, mar, tel	esp	7370	0.72	0.71
3	eng, esp, mar, tel	afr hin pan	9432	0.79 0.77 0.02	0.81 0.82 -0.00
4	arq, ary, hau	amh	3921	0.66	0.69
5	amh, arq, ary, hau	arb	4913	0.42	0.44
6	amh, ary, hau	arq	3652	0.55	0.37
7	amh, arq, hau	ary	3989	0.82	0.83
8	amh, arq, ary	hau	3117	0.68	0.66

Table 4: Model variants based on language vectors, language cell state vectors and language families. All variants include eng for training.

	amh	ary	esp	hau	kin
afr	0.75	0.57	0.83	0.79	0.82
amh	-	0.62	0.66	0.69	0.71
ary	0.62	-	0.54	0.61	0.49
arb	0.76	0.73	0.73	0.73	0.79
esp	0.66	0.54	-	0.76	0.82
hau	0.69	0.61	0.76	-	0.84
hin	0.80	0.73	0.74	0.72	0.71
ind	0.71	0.65	0.83	0.76	0.76
kin	0.71	0.49	0.82	0.84	-

Table 5: Cosine similarities between source languages (columns) and target languages (rows). Language vectors are obtained from lang2vec: **kNN** (cell_state vectors) (Malaviya et al., 2017). We exclude four languages for which we cannot obtain feature vectors: arq, mar, tel, eng.

	amh	ary	esp	hau	kin
afr	0.07	-0.05	0.23	0.07	0.22
amh	-	-0.01	0.00	0.07	0.05
ary	-0.01	-	-0.06	-0.03	0.06
arb	0.07	-0.05	0.13	-0.03	0.11
esp	0.00	-0.06	-	0.22	0.23
hau	0.07	-0.03	0.22	-	0.19
hin	0.13	-0.01	0.06	0.07	0.06
ind	0.00	0.05	0.11	0.06	0.09
kin	0.05	0.06	0.23	0.19	-

Table 6: Cosine similarities between source languages (columns) and target languages (rows). Language vectors are obtained from lang2vec: `L2V-LRN` (Littell et al., 2017). We exclude four languages for which we cannot obtain **`L2V-LRN`** features: arq, mar, tel, eng.

	amh	ary	esp	hau	kin	arq	mar	tel	eng
afr	0.86	0.70	0.76	0.87	0.85	0.73	0.80	0.80	0.82
amh	-	0.73	0.80	0.82	0.78	0.76	0.95	0.84	0.76
ary	0.73	-	0.73	0.67	0.73	0.97	0.77	0.69	0.70
arb	0.85	0.90	0.76	0.77	0.76	0.93	0.80	0.71	0.73
esp	0.80	0.73	-	0.73	0.78	0.76	0.84	0.74	0.86
hau	0.82	0.67	0.73	-	0.82	0.69	0.77	0.77	0.78
hin	0.82	0.75	0.82	0.75	0.82	0.77	0.87	0.87	0.78
ind	0.76	0.70	0.76	0.78	0.85	0.73	0.80	0.80	0.91
kin	0.78	0.73	0.78	0.82	-	0.76	0.82	0.82	0.85
arq	0.76	0.97	0.76	0.69	0.76	-	0.80	0.71	0.73
eng	0.76	0.70	0.86	0.78	0.85	0.73	0.80	0.80	-
pan	0.95	0.77	0.84	0.77	0.82	0.80	1.00	0.89	0.80

Table 7: Cosine similarities between source languages (columns) and target languages (rows). Language vectors are obtained from lang2vec: **`L2V-Phon`** (Littell et al., 2017).

	amh	ary	esp	hau	kin	arq	mar	tel	eng
afr	0.62	0.66	0.73	0.71	0.55	0.67	0.62	0.56	0.85
amh	-	0.59	0.63	0.57	0.51	0.60	0.72	0.77	0.59
ary	0.59	-	0.81	0.72	0.63	0.93	0.50	0.48	0.73
arb	0.61	0.87	0.75	0.64	0.64	0.85	0.49	0.50	0.64
esp	0.63	0.81	-	0.74	0.59	0.81	0.56	0.52	0.82
hau	0.57	0.72	0.74	-	0.65	0.78	0.52	0.34	0.75
hin	0.74	0.67	0.68	0.57	0.46	0.65	0.83	0.78	0.62
ind	0.45	0.73	0.66	0.67	0.52	0.74	0.36	0.32	0.73
kin	0.51	0.63	0.59	0.65	-	0.64	0.39	0.38	0.49
arq	0.60	0.93	0.81	0.78	0.64	-	0.49	0.47	0.74
eng	0.59	0.73	0.82	0.75	0.49	0.74	0.56	0.52	-
pan	0.71	0.68	0.70	0.59	0.49	0.67	0.79	0.75	0.61

Table 8: Cosine similarities between source languages (columns) and target languages (rows). Language vectors are obtained from lang2vec: **`L2V-Syn`** (Littell et al., 2017).

	amh	ary	esp	hau	kin	arq	mar	tel	eng
afr	0.65	0.56	0.62	0.61	0.69	0.61	0.67	0.68	0.69
amh	-	0.76	0.74	0.83	0.80	0.73	0.73	0.64	0.70
ary	0.76	-	0.62	0.70	0.70	0.85	0.63	0.57	0.65
arb	0.72	0.83	0.65	0.70	0.71	0.98	0.64	0.60	0.73
esp	0.74	0.62	-	0.67	0.68	0.64	0.66	0.66	0.64
hau	0.83	0.70	0.67	-	0.76	0.72	0.64	0.59	0.62
hin	0.66	0.69	0.57	0.62	0.69	0.77	0.72	0.77	0.71
ind	0.88	0.75	0.76	0.79	0.82	0.77	0.74	0.68	0.76
kin	0.80	0.70	0.68	0.76	-	0.72	0.65	0.63	0.69
arq	0.73	0.85	0.64	0.72	0.72	-	0.65	0.62	0.71
eng	0.70	0.65	0.64	0.62	0.69	0.71	0.76	0.67	-
pan	0.71	0.60	0.69	0.65	0.71	0.66	0.82	0.78	0.77

Table 9: Cosine similarities between source languages (columns) and target languages (rows). Language vectors are obtained from lang2vec: **L2V-Inv** (Littell et al., 2017).

	amh	ary	esp	hau	kin	arq	mar	tel	eng
afr	0.00	0.00	0.11	0.00	0.00	0.00	0.15	0.00	0.50
amh	-	0.40	0.00	0.17	0.00	0.43	0.00	0.00	0.00
ary	0.40	-	0.00	0.16	0.00	0.94	0.00	0.00	0.00
arb	0.46	0.87	0.00	0.18	0.00	0.93	0.00	0.00	0.00
esp	0.00	0.00	-	0.00	0.00	0.00	0.12	0.00	0.10
hau	0.17	0.16	0.00	-	0.00	0.17	0.00	0.00	0.00
hin	0.00	0.00	0.11	0.00	0.00	0.00	0.46	0.00	0.13
ind	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
kin	0.00	0.00	0.00	0.00	-	0.00	0.00	0.00	0.00
arq	0.43	0.94	0.00	0.17	0.00	-	0.00	0.00	0.00
eng	0.00	0.00	0.10	0.00	0.00	0.00	0.14	0.00	-
pan	0.00	0.00	0.12	0.00	0.00	0.00	0.50	0.00	0.14

Table 10: Cosine similarities between source languages (columns) and target languages (rows). Language vectors are obtained from lang2vec: **L2V-Fam** (Littell et al., 2017).

	amh	ary	esp	hau	kin	arq	mar	tel	eng
afr	0.97	0.91	0.90	0.96	0.99	0.91	0.92	0.92	0.87
amh	-	0.95	0.95	0.98	0.99	0.96	0.97	0.96	0.94
ary	0.95	-	1.00	0.98	0.94	1.00	0.88	0.87	0.99
arb	0.99	0.95	0.96	0.97	0.97	0.97	0.98	0.97	0.96
esp	0.95	1.00	-	0.98	0.94	1.00	0.90	0.89	1.00
hau	0.98	0.98	0.98	-	0.99	0.98	0.90	0.90	0.96
hin	0.97	0.89	0.91	0.90	0.94	0.91	1.00	1.00	0.91
ind	0.89	0.77	0.79	0.81	0.87	0.79	0.96	0.96	0.79
kin	0.99	0.94	0.94	0.99	-	0.95	0.94	0.94	0.92
arq	0.96	1.00	1.00	0.98	0.95	-	0.90	0.90	0.99
eng	0.94	0.99	1.00	0.96	0.92	0.99	0.90	0.89	-
pan	0.96	0.90	0.91	0.91	0.93	0.92	1.00	1.00	0.92

Table 11: Cosine similarities between source languages (columns) and target languages (rows). Language vectors are obtained from lang2vec: **L2V-Geo** (Littell et al., 2017).

	Indo-European					Afro-Asiatic					Other		
FURINA	eng	esp	afr	hin	pan	amh	arb	arq	ary	hau	ind	kin	avg
MIN	0.34	0.38	0.48	0.35	-0.19	0.68	0.04	0.00	0.28	0.33	0.22	0.23	0.36
Furina$_{eng}$	-	0.54	0.79	0.70	-0.14	0.74	0.37	0.45	0.59	0.63	0.44	0.53	0.62
Furina$_{L2V\text{-}kNN}$	-	0.62	0.71	0.35	-	0.73	0.42	-	0.28	0.64	0.42	0.68	0.53
L2V-Pho	0.76	0.56	0.79	0.77	0.03	0.76	0.46	0.48	0.63	0.43	0.43	0.68	0.63
L2V-Syn	0.76	0.56	0.80	0.78	0.03	0.74	0.39	0.48	0.63	0.54	0.34	0.68	0.63
L2V-Inv	0.78	0.38	0.79	0.76	0.03	0.76	0.46	0.48	0.63	0.43	0.22	0.23	0.60
L2V-Fam	0.78	0.64	0.80	0.78	0.03	0.76	0.46	0.48	0.63	0.485	-	-	0.65
L2V-Geo	0.76	0.47	0.79	0.78	0.03	0.73	0.04	0.53	0.63	0.64	0.30	0.23	0.58
L2V-LRN	-	0.62	0.71	0.35	-	0.76	0.42	-	0.59	0.33	0.42	0.54	0.54
MAX	0.79	0.64	0.81	0.78	0.06	0.76	0.53	0.55	0.77	0.66	0.45	0.78	0.71

Table 12: Single-source transfer results in terms of spearman correlation. The language selection is based on the cosine similarity of different typological features obtained from lang2vec (L2V). We additionally report the lower and upper bound (MIN, MAX) when choosing the worst and best possible donor language for each test language. Languages that are not covered by all L2V features are excluded from the average (eng, pan, arq, ind, kin).

	afr	amh	ary	arb	esp	hau	hin	ind	kin	arq	eng	pan
MIN (XLM-R)	esp	esp	amh	amh	ary	esp	esp	tel	arq	amh	esp	ary
MIN (Furina)	amh	esp	amh	amh	amh	esp	amh	amh	amh	amh	amh	ary
kNN	esp	kin	amh	kin	kin	kin	amh	esp	hau	-	-	-
L2V-Pho	hau	mar	arq	arq	eng	amh	mar+tel	eng	eng	ary	esp	mar
L2V-Syn	eng	tel	arq	ary	eng	arq	mar	arq	hau	ary	esp	mar
L2V-Inv	kin	hau	arq	arq	amh	amh	tel	amh	amh	ary	mar	mar
L2V-Fam	eng	arq	arq	arq	mar	amh+arq	mar	-	-	ary	mar	mar
L2V-Geo	kin	kin	arq	amh	arq	kin	mar	tel	amh	esp	esp	mar
L2V-LRN	esp	hau	kin	kin	kin	esp	amh	esp	esp	-	-	-
MAX (XLM-R)	eng	eng	eng	mar	hau	eng	eng	esp	mar	eng	mar	amh
MAX (Furina)	mar	arq	eng	eng	mar	mar	mar	ary	mar	mar	hau	kin

Table 13: Each cell shows a given test language and lang2vec (L2V) feature the closest source language used for single source transfer in Table 3 and Table 12. We further show the closest languages according to cell-state vectors obtained from a multilingual MT system (kNN) (Malaviya et al., 2017), see §3.2 for details. MIN and MAX show the source language for which best transfer and worst transfer performance is achieved.

Pair	**Sentence Pair**
esp-182	"Un hombre está saltando a una pared baja." "Un hombre está saltando a un muro bajo."
translated	"A man is jumping into a low wall." "A man is jumping into a low wall"

Table 14: An example from Spanish training dataset with its English translation, the label is 0.80.

C Translation quality.

We reviewed some machine-translated examples and noticed that subtle differences in the original language can be lost during translation. As shown in Table 14, the two translated sentences, apart from punctuation, share no differences while the label assigned is 0.8. This undoubtedly has the potential to interfere with the model's learning process for the STR task.

NLP_Team1@SSN at SemEval-2024 Task 1: Impact of language models in Sentence-BERT for Semantic Textual Relatedness in Low-resource Languages

Senthil Kumar B Aravindan Chandrabose Gokulakrishnan B Karthikraja TP

Department of Information Technology
Sri Sivasubramaniya Nadar College of Engineering
Chennai, Tamilnadu, INDIA
{*senthil*, *AravindanC*, *gokulakrishnan*2010598, *karthikraja*2010588} @*ssn.edu.in*

Abstract

Semantic Textual Relatedness (STR) will provide insight into the limitations of existing models and support ongoing work on semantic representations. Track A in Shared Task-1, provides pairs of sentences with semantic relatedness scores for 9 languages out of which 7 are low-resources. These languages are from four different language families. We developed models for 8 languages (except for Amharic) in Track A, using Sentence Transformers (SBERT) architecture, and fine-tuned them with multilingual and monolingual pre-trained language models (PLM). Our models for English (eng), Algerian Arabic (arq), and Kinyarwanda (kin) languages were ranked 12, 5, and 8 respectively. Our submissions are ranked 5^{th} among 40 submissions in Track A with an average Spearman correlation score of 0.74. However, we observed that the usage of monolingual PLMs did not guarantee better than multilingual PLMs in Marathi (mar), and Telugu (tel) languages in our case.

1 Introduction

Prior NLP work has largely focused on semantic similarity, a subset of relatedness, because of a lack of relatedness datasets. The first dataset for Semantic Textual Relatedness, STR-2022 was introduced by Abdalla et al., 2023, which has 5,500 English sentence pairs manually annotated using a comparative annotation framework, resulting in fine-grained scores. The semantic relatedness of two units of language is the degree to which they are close in terms of their meaning (Mohammad and Hirst, 2012). The linguistic units can be words, phrases, sentences, etc.

The most semantic similarity datasets were annotated using coarse rating labels such as integer values between 1 and 5 representing coarse degrees of closeness. These datasets suffer from issues arising due to the fixed granularity which intuits fuzzy boundaries between related and unrelated notions. The following subsection describes the difference between similarity and relatedness which is crucial in understanding the textual semantics.

1.1 Similarity versus Relatedness

As discussed in Abdalla et al., 2023, the following are the characteristics of similarity versus relatedness:

1. Two terms are considered semantically similar if there is a synonymy, hyponymy, or troponymy relation between them whereas for semantic relatedness, it's enough to have any lexical semantic relation at all between them. (example: money-cost is related whereas price-cost is similar)

2. All similar pairs are also related, but not all related pairs are similar. For example, `surgeon-scalpel`, and `tree-shade` are related, but not similar.

3. If units are sentences, then the similarity between sentence pairs exhibits paraphrase or entailment property whereas the relatedness does not support that property since it accounts for all of the commonalities that can exist between two sentences.

The analysis showed that the presence of proper nouns (PROPN), nouns, and other coarse-grained POS categories in a sentence pair impact semantic relatedness much more than any other POS. We evaluated the semantic textual relatedness of 8 languages (Algerian Arabic (arq), Moroccan Arabic (ary), Kinyarwanda (kin), Hausa (hau), Marathi (mar), Telugu (tel), English (eng) and Spanish (esp)) in Track A of SemEval Task 1: Semantic Textual Relatedness for African and Asian Languages (Ousidhoum et al., 2024b).

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1854–1859

2 Related Work

Similarity task is originally proposed to mimic human perception of the similarity level between word or sentence pairs. The first, word similarity dataset was collected in Rubenstein and Goodenough (1965), which consisted of 65-word pairs with human annotations. In general, the datasets consist of pairs of words (w1, w2) (or sentences) and human-annotated similarity scores S_h.

Abdalla et al. (2023) measured the semantic relatedness using Contextual versus Static embeddings and Unsupervised versus Supervised approach to sentence representation. In an unsupervised approach, the embedding of a sentence is derived from that of its constituent tokens. They used Word2Vec, GLoVe, and Fasttext static embeddings in unsupervised settings and the majority of the static embedding models failed to obtain better correlations with human annotation scores. The contextual embeddings from BERT and RoBERTa do not perform better than the Word2vec embeddings.

Finally, the supervised approach by finetuning the SBERT with the STR-2022 dataset captured high semantic relatedness and the Spearman correlation is 0.82 and 0.83 for BERT-based and RoBERTa-based respectively. The supervised approach using the SBERT framework by formulating a regression task leads to a better correlation score of 0.20 than the unsupervised approach.

This motivated us to use the SBERT framework to score the semantic relatedness between the pairs of sentences across 6 low-resource languages and English, and Spanish in the Track A dataset. We used 2 multilingual pre-trained language models ($LABSE, pp-mpnet-v2$) and language-specific monolingual LM for each of the languages. The following subsections describe the reason behind the selection of particular pre-trained LMs that are used in our models.

2.1 LaBSE

Multilingual pre-trained models such as mBERT (Devlin et al., 2019) and XLM-R (CONNEAU and Lample, 2019) have led to exceptional gains across a variety of cross-lingual natural language processing tasks. However, without a sentence-level objective, they do not directly produce good sentence embeddings.

Language-agnostic BERT Sentence Embedding (Feng et al., 2022) is a multilingual BERT embedding model, called LaBSE, that produces language-agnostic cross-lingual sentence embeddings for 109 languages. The model is trained on 17 billion monolingual sentences and 6 billion bilingual sentence pairs using MLM and TLM pre-training, resulting in a model that is effective even on low-resource languages for which there is no data available during training.

It was trained on parallel sentence pairs from 109 languages using a Siamese network based on the BERT architecture. The model's ability to support 109 languages makes it a powerful tool for multilingual applications and cross-lingual natural language processing tasks. This multilingual PLM is used across all the 8 models in our experiment.

PLM Type	Language Model
Monolingual	MahaSBERT, TeluguSBERT
	DziriBERT
Multilingual	Sentence-T5, LaBSE
	AfroXLMR, IndicSBERT
	pp-mpnet-v2

Table 1: Types of pre-trained LM

2.2 paraphrase-multilingual-mpnet-base-v2

This is based on the multi-lingual model of `paraphrase-mpnet-base-v2`, extended to 50+ languages by Reimers and Gurevych 2020. It uses a multilingual knowledge distillation method that allows extending existing sentence embedding models to new languages. It has achieved state-of-the-art performance on the paraphrase identification task on several benchmark datasets.

2.3 AfroXLMR

Alabi et al. (2022) proposed multilingual adaptive fine-tuning (MAFT) as a method for simultaneously adapting multilingual pre-trained language models (PLMs) on 17 of Africa's most resourced languages and three other high-resource languages widely spoken on the African continent to encourage cross-lingual transfer learning. This approach was more competitive than the AfriBERTa (Ogueji et al., 2021) pre-trained LM on various NLP tasks. We used this pre-trained LM for Kinyarwanda (kin) and Hausa (hau) languages.

2.4 IndicSBERT

The IndicSBERT exhibits strong cross-lingual capabilities and performs significantly better than

Pre-trained LM	English	Spanish
LaBSE	0.802	**0.68**
pp-mpnet-v2	0.805	0.63
sentence-t5-large	**0.824**	-
sentence-similarity-spanish-es	-	0.66

Table 2: Evaluation of Indo-European languages during development

Pre-trained LM	Marathi	Telugu
LaBSE	0.82	0.797
pp-mpnet-v2	0.77	0.747
IndicSBERT	0.58	0.61
MahaSBERT	**0.84**	-
TeluguSBERT	-	**0.811**

Table 3: Evaluation of Marathi, Telugu languages during development

alternatives like LaBSE, LASER, and paraphrase-multilingual-mpnet-base-v2 on Indic cross-lingual and monolingual sentence similarity tasks.

The authors Deode et al. (2023) proposed a simple strategy to train cross-lingual sentence representations using a pre-trained multilingual BERT model and synthetic NLI/STS data. This is the first multilingual SBERT model trained specifically for Indian languages. However, monolingual models are typically found to be performing better than multilingual ones. Hence publicly released monolingual SBERT models for 10 Indic languages. We used `MahaSBERT` for Marathi(mar), and `TeluguSBERT` for Telugu(tel) in evaluating the STR score in Track A.

2.5 DziriBERT

The Algerian dialect is mainly inspired by standard Arabic but also from Tamazight, French, Turkish, Spanish, Italian, and English. Thus the Algerian dialect has several specificities that make the use of Arabic or multilingual models inappropriate. To address this issue the authors (Abdaoui et al., 2022) collected more than one million Algerian tweets and pre-trained the first Algerian language model: DziriBERT.

DziriBERT is a BERT-based model for the Algerian dialect which was trained using the Masked Language Modeling (MLM) task. It handles Algerian text contents written using both Arabic and Latin characters. We used this model for evaluating the semantic relatedness score for the Semitic languages group - Algerian Arabic(arq), and Moroccan Arabic(ary).

3 System Overview

Given a human-annotated dataset for semantic textual relatedness, the participants are allowed to submit systems that have been trained using the labeled training datasets. Apart from that, the participating teams are also allowed to use any other publicly available datasets. We restrict the use of only the dataset provided by the task organizers so that the impact of pre-trained language models on Sentence Transformers can be analyzed for the semantic relatedness task across different low-resource languages. We used the plain vanilla SBERT architecture for fine-tuning with pre-trained LMs for text processing.

In our experiment, predicting semantic relatedness is treated as a regression task, where each sentence is represented as a vector. We use the cosine similarity between the vectors to predict their semantic relatedness, S_p, the Spearman score predicted by the system. Finally, the correlation between S_h, the Spearman score manually annotated by humans, and S_p is computed, and a higher correlation suggests good alignment with human annotations and a better embedding model. Usually, the Spearman correlation between the prediction and gold relatedness scores is used to measure the goodness of the relatedness predictions.

3.1 Dataset

The authors Ousidhoum et al. (2024a) presented SemRel2024 dataset - the first benchmark on semantic distance (similarity or relatedness) that includes low-resource African and Asian languages from five different language families. We used the sentence pairs of 8 languages from the dataset for Track A. Refer Ousidhoum et al. (2024b) to the dataset split size for training, development, and test instances for Track A. The dataset contains semantic relatedness scores for each of the pairs of sentences of 8 languages.

3.2 Training and Testing

During the development phase, only the training and development datasets are given to construct the model for each language. The training data is used to fine-tune the model and development data is used to evaluate the model performance. We report the results using the default hyperparameters

set in the sentence transformer. The PLMs are fine-tuned on training data using cosine similarity loss with batch size as 8, and number of epochs as 20. The official evaluation metric is the Spearman correlation between the predicted similarity scores and the human-annotated gold scores.

During the test phase, we combined the training data + development data to fine-tune the model, and the unseen test data was used to predict the semantic relatedness score. Models using various pre-trained LMs are evaluated using Spearman correlation during the development phase. The model with the maximum Spearman correlation score is used during the testing phase to submit our results. The table 1 lists the types of pre-trained LMs and the corresponding LMs used in our study.

4 Experimental Setup

We aim to focus on the impact of the Sentence-BERT deep neural network in semantic textual relatedness scoring tasks, and the benefit of multilingual/monolingual pre-trained LMs over the task especially for the low-resource languages.

4.1 SentenceBERT

Unlike BERT, SentenceTransformer or SBERT by Reimers and Gurevych (2020) uses a Siamese architecture, where it contains two BERT architectures that are essentially identical and share the same weights. It processes two sentences as pairs during training. This neural network architecture is appropriate for pair-wise semantic sentence tasks such as Sentence Textual Similarity (STS), Semantic Textual Relatedness (STR), Natural Language Inference (NLI), and paraphrase identification tasks. This network leverages the two BERT architectures in parallel to compute/score the similarity/relatedness of pair-wise sentences.

Consider a pair of sentences S1 and S2 that are to be fed into the network. Feed a sentence S1 to BERT A and S2 to BERT B in the SBERT network. Each BERT outputs pooled sentence embeddings u and v respectively. The cosine similarity between these two embeddings (u, v) is computed by using mean-squared error loss as the objective function. This outputs the regressive score between 0 to 1. This is the predicted semantic relatedness score by the model between a pair of sentences S1 and S2. We developed all the models using SBERT for each of the 8 languages (except for Amharic) in Track A.

Pre-trained LM	Algerian Arabic	Moroccan Arabic
LaBSE	0.58	**0.799**
pp-mpnet-v2	0.53	0.73
DziriBERT	**0.67**	0.64

Table 4: Evaluation of Semitic languages during development

Pre-trained LM	Kinyarwanda	Hausa
LaBSE	0.579	0.715
pp-mpnet-v2	0.58	0.67
AfroXLMR	**0.61**	**0.73**

Table 5: Evaluation of African languages during development

4.2 Evaluation during development phase

The train ad development split data for each of the languages are as mentioned in the Ousidhoum et al. (2024a). We used two multilingual pre-trained LMs: LaBSE[1] and paraphrase-multilingual-mpnet-base-v2[2] (in short `pp-mpnet-v2`) across all the models. The idea behind using multilingual PLM for all 8 languages is primarily to check the performance of MLM for semantic textual relatedness tasks in low-resource languages. Apart from that, language-specific monolingual pre-trained LMs are also used in each of the models. During the development phase, the model that scored the maximum Spearman correlation is selected and applied during the testing phase. The models developed for each of the languages along with the pre-trained LMs used and its score are discussed below.

The table 2 shows that `sentence-t5-large` [3] (Ni et al., 2022), a text-to-text model showed better performance for the `English` language. The model using `LaBSE` scored higher than the other multilingual and monolingual LM for `Spanish` during evaluation in the development phase.

Table 3 shows that the monolingual models such as `MahaSBERT`[4] and `TeluguSBERT`[5] perform well than the multilingual models. The interesting fact to note is that even the `IndicSBERT`[6] - one of the popular multilingual models pre-trained on 14 Indian languages, scored poorly than the

[1] sentence-transformers/LaBSE

[2] sentence-transformers/paraphrase-multilingual-mpnet-base-v2

[3] sentence-transformers/sentence-t5-large

[4] l3cube-pune/marathi-sentence-similarity-sbert

[5] l3cube-pune/telugu-sentence-similarity-sbert

[6] ai4bharat/indic-bert

Language	Model	Predict	Rank	baseline	LM type	diff.
English (eng)	SBERT-T5	**0.8352**	12	0.83	**MultiLM**	+0.0052
Spanish (esp)	SBERT-LaBSE	**0.7045**	9	0.7	**MultiLM**	+0.0045
Marathi (mar)	SBERT-MahaSBERT	0.8711	10	0.88	MonoLM	-0.0089
Telugu (tel)	SBERT-TeluguSBERT	0.7889	17	0.82	MonoLM	-.0.0311
Algerian Arabic (arq)	SBERT-DziriBERT	**0.6226**	5	0.6	**MonoLM**	+0.0226
Moroccan Arabic (ary)	SBERT-LaBSE	0.7446	16	0.77	MultiLM	-0.0254
Kinyarwanda (kin)	SBERT-AfroXLMR	**0.7233**	8	0.72	**MultiLM**	+0.0033
Hausa (hau)	SBERT-AfroXLMR	0.6281	11	0.69	MultiLM	-0.0619

Table 6: Evaluation of our SBERT-based models during the test phase. Boldface highlights the score more or equal to the baseline

LaBSE and pp-mpnet-v2 generic multilingual LM. IndicSBERT is one of the regional multilingual LMs trained in Indian languages.

Similarly for the Semitic languages such as Algerian Arabic and Moroccan Arabic, DziriBERT[7] PLM performed better than the generic multilingual LM in the Algerian Arabic language as shown in Table 4. The DziriBERT was specifically pre-trained on Algerian dialects. For Moroccan Arabic, a model with LaBSE had a better score than the model using DziriBERT LM. As per our knowledge, we do not find any monolingual pre-trained LM for Moroccan Arabic that improves the score than the LaBSE. This is one of the major drawbacks of low-resource languages. The availability of good pre-trained LM for task-specific or generic purposes is scarce in low-resource languages.

For African languages, the performance of the model using AfroXLMR[8] pre-trained LM scored better than the other models using generic pre-trained LMs as shown in table 5. This indicates that the use of appropriate pre-trained LMs is more important for semantic relatedness tasks than the generic pre-trained multilingual language models.

5 Result

During the testing phase, we combined the training + development data as training data to fine-tune the model that yielded the maximum score during the development phase. Then the model is tested with the test dataset of the corresponding language. The predicted sentence relatedness score by the models is submitted as a result and is evaluated using the Spearman coefficient. The results are shown in the Table 6. It is evident from the table 6 that almost 4 models had reached a score equal to or above the baseline score which is highlighted using boldface. The difference between the baseline and the model prediction is indicated with the + and - sign. The difference in Spearman correlation value with '+' indicates the improvement whereas the '-' sign indicates the poor performance of the model.

SBERT-based models for the languages English (eng), Algerian Arabic (arq), and Kinyarwanda (kin) performed more than the baseline Spearman score. SBERT-LaBSE model for Spanish (esp) scored almost equal to the baseline system. Even though the monolingual models SBERT-MahaSBERT for Marathi and SBERT-TeluguSBERT for Telugu showed better performance during the development phase, failed to score above the baseline during testing in respective languages. Similarly, SBERT-based models trained using multilingual pre-trained LM for Moroccan Arabic (ary) and Hausa (hau) languages scored lesser than the baseline model in the test phase.

5.1 Conclusion

Table 6 depicts the impact of pre-trained language models (LM) in SBERT for the various low-resource languages. The usage of monolingual LM in Marathi (mar) and Telugu (tel) did not guarantee a greater performance than the baseline system. This shows the limitations of existing state-of-the-art monolingual pre-trained LM MahaSBERT, TeluguSBERT for the STR task.

Apart from that, the multilingual pre-trained LM such as LaBSE, AfroXLMR did not perform well for Moroccan Arabic (ary) and Hausa (hau) which are from Afro-Asiatic language family. This shows the existence of poor resources such as pre-trained LM in those languages. By default, the monolingual LM did not guarantee better performance than the multilingual pre-trained LM, especially for the low-resource languages.

[7] alger-ia/dziribert

[8] Davlan/afro-xlmr-large

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Amine Abdaoui, Mohamed Berrimi, Mourad Oussalah, and Abdelouahab Moussaoui. 2022. Dziribert: a pre-trained language model for the algerian dialect.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Samruddhi Deode, Janhavi Gadre, Aditi Kajale, Ananya Joshi, and Raviraj Joshi. 2023. L3Cube-IndicSBERT: A simple approach for learning cross-lingual sentence representations using multilingual BERT. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 154–163, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Saif M. Mohammad and Graeme Hirst. 2012. Distributional measures of semantic distance: A survey.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024a. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Meriem Beloucif, Christine De Kock, Oumaima Hourrane, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Krishnapriya Vishnubhotla, Seid Muhie Yimam, and Saif M. Mohammad. 2024b. SemEval-2024 task 1: Semantic textual relatedness for african and asian languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633.

ShefCDTeam at SemEval-2024 Task 4: A Text-to-Text Model for Multi-Label Classification

Meredith Gibbons[1], Maggie Mi[1], Aline Villavicencio[1,2] Xingyi Song[1]
[1] Department of Computer Science, The University of Sheffield, UK
[2] Institute of Data Science and Artificial Intelligence, University of Exeter, UK
{magibbons1, zmi1, x.song}@sheffield.ac.uk
a.villavicencio@exeter.ac.uk

Abstract

This paper presents our findings for SemEval-2024 Task 4. We submit only to subtask 1, applying the text-to-text framework using a FLAN-T5 model with a combination of parameter efficient fine-tuning methods - low-rank adaptation and prompt tuning. Overall, we find that the system performs well in English, but performance is limited in Bulgarian, North Macedonian and Arabic. Our analysis raises interesting questions about the effects of label order and label names when applying the text-to-text framework.

1 Introduction

Social media platforms have become increasingly popular over time (Perrin, 2015). Whilst this enables greater public discourse, information and disinformation can also be presented purposefully to influence opinions online . Therefore, it is important to explore the detection of persuasion techniques. By fulfilling this goal, strategies that counteract false or misleading narratives can developed, and internet users can be empowered to think more critically about what they see online.

This paper describes our submission for SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. We took a text only approach, and as such we only tackled subtask 1 - given only the "textual content" of a meme, our system must identify which persuasion techniques (of a possible 20) are used (Dimitrov et al., 2024). The labels are organized in a hierarchy (see figure 1) and multiple labels may apply to the same data point. For example:

> **Text:** HISTORY HAS SHOWN THAT THESE ARE THE FIRST TWO THINGS BANNED\\n\\nBY TOTALITARIAN GOVERNMENTS
>
> **Labels**: Loaded Language, Thought-terminating cliché

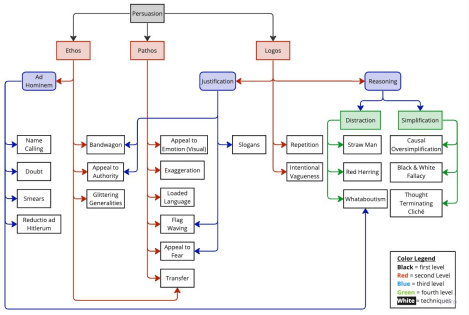


Figure 1: The hierarchical structure of the labels (Dimitrov et al., 2024).

In recognition of the diverse and intriguing use of language for manipulative communication, we target our exploration using a transformer-based architecture due to the ability of such models to capture linguistic intricacies (Plaza-del arco et al., 2023; Tenney et al., 2019). Specifically, we investigate this task using the text-to-text model FLAN-T5 (Chung et al., 2022).

2 Background

Research on identifying persuasion techniques in memes builds on the efforts of propaganda detection (Da San Martino et al., 2021; Dimitrov et al., 2021). Rashkin et al. (2017) trained models using n-gram TF-IDF feature vectors on a four category news reliability classification task. Barrón-Cedeño et al. (2019) both replicated the work of Rashkin et al. (2017) and applied n-grams to propaganda detection under binary classification. More recently, Da San Martino et al. (2019) took a more fine-grained approach. They developed a dataset of news articles with an annotation schema consisting of 18 propaganda techniques. They proposed a multi-granularity network using contextual embeddings derived with BERT (see also Da San Martino et al., 2020). Piskorski et al. (2023) presents a multilingual and multifaceted dataset of news articles, annotated with genre, framing and persuasion tech-

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1860–1867

niques. They also evaluated the performance of a transformer model at various granularity levels - token-level, sentence-level, paragraph-level, and document-level.

To the best of our knowledge, there has been no work completed on exploring text-to-text (also known as sequence-to-sequence, or Seq2Seq) models for this multilingual, multi-label classification task in the domain of meme language. Text-to-text models take in text as input and output new text. Models such as T5 can be applied to many different tasks under the text-to-text framework (Raffel et al., 2019). They have also been shown to be effective in zero-shot settings (Chung et al., 2022; Plaza-del arco et al., 2023).

3 System Overview

We use FLAN-T5 (Chung et al., 2022) as our base model. FLAN-T5 was created by fine-tuning T5 (Raffel et al., 2019) on a mixture of tasks including text classification, question answering, and translation. The model regards every task as a text-to-text task.

We train in two steps:

1. LoRA; Low-Rank Adaptation (Hu et al., 2021)

2. Prompt Tuning (Lester et al., 2021)

For both steps all of the original FLAN-T5 parameters are frozen, lessening training time and hardware requirements. As both methods introduce their own set of distinct parameters, the LoRA parameters do not need to be trainable during prompt tuning. We first train using LoRA, then freeze the values of the introduced LoRA parameters and train using prompt tuning to produce the final model.

3.1 LoRA

Neural networks contain many dense layers, which transform input x to output h via matrix multiplication. Without model adaption, the pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$ produces output as follows:

$$h = W_0 x$$

After model adaptation, the updated output can be represented as follows:

$$h_{adapted} = W_0 x + \Delta W x$$

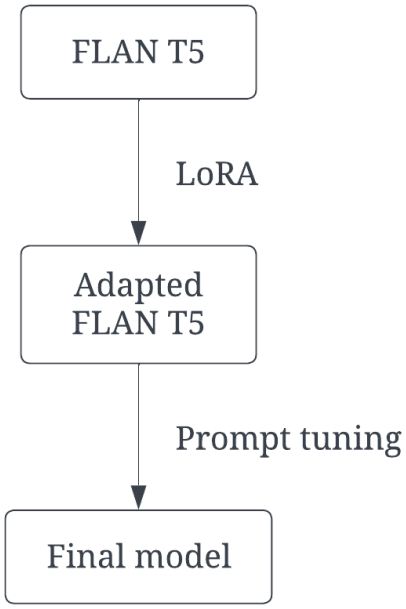


Figure 2: Training steps for our model.

where ΔW is the overall change to the weights, optimised during training. LoRA constrains ΔW by decomposing it into two low-rank matrices, $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where $r << min(d, k)$:

$$h_{LoRA} = W_0 x + BAx$$

This process is summarised in figure 3. A and B are trainable parameters, initialised as a random Gaussian and 0 respectively to give an initial $BA = \Delta W$ of 0. $\Delta W x$ is scaled by $\frac{\alpha}{r}$, where α is a hyperparameter. Hu et al. (2021) applied LoRA to attention weights, achieving on par or better performance than full fine-tuning with only a fraction of the trainable parameters.

3.2 Prompt Tuning

In prompt engineering, a "hard prompt" is prepended to the input and used to guide the model to produce the desired output. Prompt tuning instead learns a "soft prompt", wherein the prompt tokens are taken as learnable parameters.

For input consisting of a token sequence $x_0, x_1, ..., x_n$, the tokens are first transformed to the embedding $X_e \in \mathbb{R}^{n \times e}$, where e is the dimension of the embedding space. The soft prompt, $P_e \in \mathbb{R}^{p \times e}$, where p is the length of the prompt, is concatenated to X_e to form new input matrix

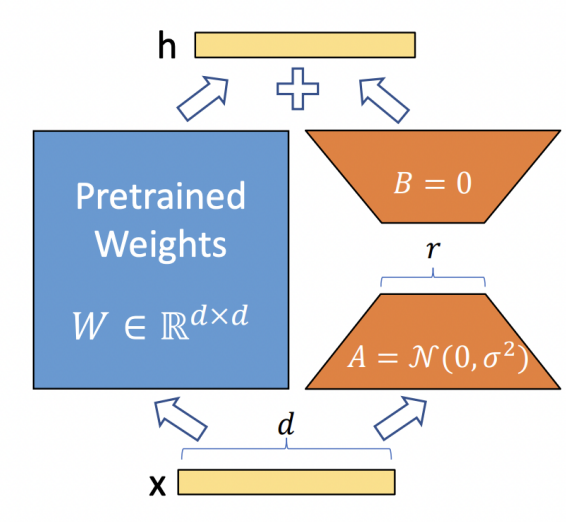


Figure 3: Overview of the LoRA method (Hu et al., 2021).

$[P_e; X_e] \in \mathbb{R}^{(p+n)\times e}$. During training, all model parameters are frozen and only P_e is optimised.

This method drastically reduces the number of required parameters, while achieving comparable performance to full fine-tuning when applied to very large models.

4 Experimental Setup

For hardware reasons, we use a sharded version of FLAN-T5-XXL[1] loaded in 8-bit precision.

The training set (size = 7000) was used for the LoRA training and the validation set (size = 500) was used for the prompt tuning.

Preprocessing was required to transform the data into an appropriate format for text-to-text training. We transform the input text to lower case, and for LoRA we prepended a simple task prompt. For example:

> NEW POLL\\n\\n82 percent of voters support TERM LIMITS ON CONGRESS\\n

becomes

> which persuasion techniques are in this text? text: new poll\\n\\n82 percent of voters support term limits on congress\\n

When preprocessing the labels, we observed that many original labels were metaphorical and/or lengthy, such as 'Glittering generalities (Virtue)'. Theorising that these sequences would be more difficult for the model to generate, we replace each label with a simplified (if applicable), lower case version. Finally, we concatenate the labels into a comma-separated list. Some examples are listed in table 1 - see Appendix A for a full list of simplified labels.

Original	Preprocessed
['Bandwagon']	'appeal to popularity'
['Repetition', 'Name calling/Labeling']	'repetition, labeling'
[]	'none'

Table 1: Examples of preprocessed labels for text-to-text training.

We use the PEFT implementation of LoRA and prompt tuning (Mangrulkar et al., 2022). For LoRA, we train for 5 epochs with a learning rate of 0.001. We mostly use the same hyperparameters for prompt tuning as Mozes et al. (2023) on T5-XXL. We initialise the prompt as:

> 'which persuasion techniques are in this text? text: '

More details on hyperparameters for both training steps can be found in Appendix B.

The evaluation measure used in this task is hierarchical F1 (Kiritchenko et al., 2006), which takes into account the tree structure of the labels when calculating model performance.

5 Results

Our final results are summarised in table 2[2]. Our English language result places us slightly above the centre of the leaderboard. Our Bulgarian result places lower, but is still superior to the baseline. Our North Macedonian result is below baseline performance. While FLAN-T5 was fine-tuned on a small number of Bulgarian language tasks during training, no North Macedonian language tasks were included. Likely due to the absence of Bulgarian and North Macedonian data in our training data and the small size of the corresponding test sets (size = 436 and 259 respectively), our results on these languages are much more variable than our English results.

[1] https://huggingface.co/philschmid/flan-t5-xxl-sharded-fp16

[2] All reported results obtained after the original task deadline.

	Hierarchical Precision
English	0.6701 ± 0.0025
Bulgarian	0.4631 ± 0.0069
N. Macedonian	0.4804 ± 0.0007
	Hierarchical Recall
English	0.6142 ± 0.0057
Bulgarian	0.2575 ± 0.0307
N. Macedonian	0.1882 ± 0.0160
	Hierarchical F1
English	0.6409 ± 0.0020
Bulgarian	0.3302 ± 0.0271
N. Macedonian	0.2700 ± 0.0164

Table 2: Hierarchical precision, recall, and F1 for our model on the test sets; average and range across two repeats.

The model failed to generalize to the fourth language, Arabic, despite its presence in the FLAN-T5 training data - we did not make a submission for this language as the model predicted no labels for all inputs.

5.1 Error Analysis

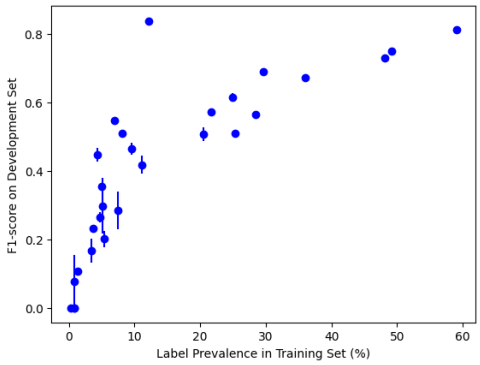


Figure 4: Prevalence of each label in the training set versus average F1 score on the English development set (size = 1000) over two repeats. Error bars show the range of values[3].

To investigate the errors of our model, we analysed the data on a per-label basis using our best performing language, English. Instead of using hierarchical F1, we split the multilabel task into 20 binary tasks (one for the prediction of each label) and calculated the average F1 score for each. In general, our system performed better on labels that were common in the training data (see figure 4). Several labels with very low training set prevalence had F1 scores of zero.

A notable result was the label 'Appeal to authority', which achieved a very high average F1 score of 0.838 while appearing in only 12.14% of the training data. Most data labelled with 'Appeal to authority' contains a quote, leading to the simplification of the label to 'quoting'. This clear pattern may have contributed to the higher average F1 score.

Other than 'Appeal to authority', the highest performing labels were non-leaf labels such as 'Ethos'[4]. These categories are very prevalent in the training data, so higher F1 scores are expected.

5.2 Further Analysis

We investigated two features of our system which may have affected the performance:

1. Ordered labels
2. Simplified label names

5.2.1 Ordered Labels

The text-to-text format necessitates that the labels be placed in an order (see table 1). This trains the model to associate an order with the labels - however, the order that the labels appear holds no semantic significance. For instance, "smears, slogans" is equivalent to "slogans, smears". In the data, there is a bias in the lists of labels in which certain labels ('Appeal to authority', 'Loaded Language', and 'Doubt') usually occur at the start. Labels such as 'Smears' usually occur at the end of the list, although the bias is not as strong as that of 'Appeal to authority'. Therefore, superfluous information may have been introduced to the model, decreasing the performance.

Alternatively, the model may leverage label order to reduce the number of possibilities while decoding, improving the performance. The typical positioning of 'Appeal to authority' at the start of the label list is another factor that may have made it an easier label to predict.

To investigate the effect of label order, we trained a separate version of our model, in which the labels of the training and validation sets (used for LoRA and prompt tuning respectively) were randomly shuffled. Our results are outlined in table 3[5], showing a slight increase in English hierarchical F1 and a much greater increase for Bulgarian and North Macedonian. This suggests that the bias in the label order may be detrimental to overall performance.

[3]As the range of F1 scores for some labels was zero or close to zero, not all error bars are visible.

[4]The model does not predict these labels directly. For the error analysis, the ancestor labels of each predicted label were added to the prediction in post-processing.

[5]All reported results obtained after the original task deadline.

	Hierarchical Precision
English	0.6978 ± 0.0031
Bulgarian	0.4362 ± 0.0117
N. Macedonian	0.4355 ± 0.0081
	Hierarchical Recall
English	0.6037 ± 0.0039
Bulgarian	0.3443 ± 0.0218
N. Macedonian	0.2724 ± 0.0228
	Hierarchical F1
English	0.6473 ± 0.0036
Bulgarian	0.3847 ± 0.0181
N. Macedonian	0.3349 ± 0.0196

Table 3: Hierarchical precision, recall, and F1 on the test sets for our model trained using shuffled labels; average and range across two repeats.

5.2.2 Simplified Label Names

Simplified labels (see Appendix A) were manually determined and focused on semantic simplicity and length. Despite this, many simplified labels were long in order to convey the concept of the persuasion technique, and some labels could not be easily simplified, being left with metaphorical or vague meanings.

To investigate the effect of the label names on performance, we compared the simplified label names with the per-label F1 scores. Table 4 shows the average per-label F1 score for the English development set and the prevalence of each label in the training set. As is also shown in figure 4, there is a correlation between average F1 score and training set prevalence. However, there are exceptions - 'virtue', the simplification of 'Glittering generalities (Virtue)', is a short and semantically obvious label and performs better than expected. Meanwhile, the longer and more metaphorical 'black and white thinking' has a lower average F1 score than expected.

This evidence suggests that longer and more complex labels may compromise text-to-text model performance, but more study is needed to reach a definitive conclusion. For example, the unusually high performance of 'quoting' is likely influenced by other factors. Some persuasion techniques may be easier or harder to detect regardless of label name.

6 Conclusion

In this paper we present a case study for the application of the text-to-text framework to multilabel classification. While our model exhibits some strengths, it did not achieve performance on par with top-ranking results. However, our analysis shows the potential for label names to affect performance, and suggests that shuffling labels during training may lead to increased performance.

Simplified Label	F1	Prevalence (%)
quoting	0.838	12.14
loaded language	0.616	25.00
labeling	0.574	21.69
smears	0.564	28.43
virtue	0.547	6.97
appeal to identity	0.509	8.16
slogans	0.464	9.53
repetition	0.447	4.36
black and white thinking	0.418	11.14
doubt	0.355	5.00
exaggeration or minimisation	0.298	5.09
shutting down discussion	0.285	7.54
appeal to fear or prejudice	0.265	4.81
whataboutism	0.232	3.69
causal oversimplification	0.167	3.43
appeal to popularity	0.108	1.39
guilt by association	0.077	0.90
straw man	0.000	0.89
red herring	0.000	0.84
obfuscation	0.000	0.30

Table 4: Simplified label names, average F1 score on the English development set over two repeats, and the prevalence of each label in the training set. The labels are ordered by average F1 score.

Limitations

Our paper has several limitations. Firstly, we only report results for our model across two repeats. This means that by chance, our results may appear to be better or worse than they would be on average. We only use English training data, which likely led to lower performance on the Bulgarian, North Macedonian, and Arabic test sets. Finally, we did not use a full-precision version of FLAN-T5-XXL due to hardware concerns. This likely led to decreased performance across all languages.

Acknowledgements

This work was supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [grant number EP/S023062/1]. We also wish to thank the reviewers for their feedback and efforts.

References

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2021. A survey on computational propaganda detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Giovanni Da San Martino, Shaden Shaar, Yifan Zhang, Seunghak Yu, Alberto Barrón-Cedeño, and Preslav Nakov. 2020. Prta: A system to support the analysis of propaganda techniques in the news. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 287–293, Online. Association for Computational Linguistics.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Detecting propaganda techniques in memes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6603–6617, Online. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A. Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: application to text categorization. In *Proceedings of the 19th International Conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence*, AI'06, page 395–406, Berlin, Heidelberg. Springer-Verlag.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Maximilian Mozes, Jessica Hoffmann, Katrin Tomanek, Muhamed Kouate, Nithum Thain, Ann Yuan, Tolga Bolukbasi, and Lucas Dixon. 2023. Towards agile text classifiers for everyone.

A. Perrin. 2015. *Social Media Usage: 2005-2015: 65% of Adults Now Use Social Networking Sites–a Nearly Tenfold Jump in the Past Decade*. Pew Research Trust.

Jakub Piskorski, Nicolas Stefanovitch, Nikolaos Nikolaidis, Giovanni Da San Martino, and Preslav Nakov. 2023. Multilingual multifaceted understanding of online news in terms of genre, framing, and persuasion techniques. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3001–3022, Toronto, Canada. Association for Computational Linguistics.

Flor Miriam Plaza-del arco, Debora Nozza, and Dirk Hovy. 2023. Respectful or toxic? using zero-shot learning with language models to detect hate speech. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 60–68, Toronto, Canada. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

A Simplified Labels

This appendix contains the simplified labels used in preprocessing. We did not remove all metaphorical references, leaving those which are relatively common (e.g. 'red herring') as FLAN-T5 is likely to have encountered them during training. As 'whataboutism' is difficult to explain succinctly, we left it as-is. All simplified labels are listed in table 5.

B Training Hyperparameters

Table 6 shows the training hyperparameters used in LoRA and prompt tuning. For our final output, we limit the length of the generated text to 20 tokens.

	Hyperparameter	**Value**
LoRA	Epochs	5
	Learning Rate	0.001
	Rank	16
	α	32
	Dropout	0.05
	Target modules	q,v
Prompt Tuning	Epochs	1
	Learning Rate	0.1
	Weight decay	0.00001
	Batch size	32
	Prompt tokens	10

Table 6: Hyperparameters used in LoRA training and prompt tuning.

Original Labels	Simplified Labels
Black-and-white Fallacy/Dictatorship	black and white thinking
Loaded Language	loaded language
Glittering generalities (Virtue)	virtue
Thought-terminating cliché	shutting down discussion
Whataboutism	whataboutism
Slogans	slogans
Causal Oversimplification	causal oversimplification
Smears	smears
Name calling/Labeling	labeling
Appeal to authority	quoting
Exaggeration/Minimisation	exaggeration or minimisation
Repetition	repetition
Flag-waving	appeal to identity
Appeal to fear/prejudice	appeal to fear or prejudice
Reductio ad hitlerum	guilt by association
Doubt	doubt
Misrepresentation of Someone's Position (Straw Man)	straw man
Obfuscation, Intentional vagueness, Confusion	obfuscation
Bandwagon	appeal to popularity
Presenting Irrelevant Data (Red Herring)	red herring

Table 5: Labels before and after simplification.

NLPNCHU at SemEval-2024 Task 4: A Comparison of MDHC Strategy and In-domain Pre-training for Multilingual Detection of Persuasion Techniques in Memes

Shih-Wei Guo[1], **Yu-Ting Lin**[2], **Yu-An Lu**[3],**Yao-Chung Fan**[1*]
[1]Department of Computer Science and Engineering,
National Chung Hsing University, Taiwan
[2]Taipei Municipal Chenggong High School, Taiwan
[3]National Chupei Senior High School, Taiwan
{cometlcc,dong1214.mailbox,luyuam0}@gmail.com, yfan@nchu.edu.tw

Abstract

This study presents a systematic method for identifying 22 persuasive techniques used in multilingual memes. We explored various fine-tuning techniques and classification strategies, such as data augmentation, problem transformation, and hierarchical multi-label classification strategies. Identifying persuasive techniques in memes involves a multimodal task. We fine-tuned the XLM-RoBERTA-large-twitter language model[1], focusing on domain-specific language modeling, and integrated it with the CLIP visual model's embedding to consider image and text features simultaneously. In our experiments, we evaluated the effectiveness of our approach by using official validation data in English. Our system in the competition, achieving competitive rankings in Subtask1 and Subtask2b across four languages: English, Bulgarian, North Macedonian, and Arabic. Significantly, we achieved 2nd place ranking for Arabic language in Subtask 1.

1 Introduction

Propaganda and advertising serve as examples of persuasive discourse, which aims to change another's behavior, feelings, intentions, or views through communication, often in a one-sided manner (Lakoff, 1982). Hence, the context in which the communication occurs is crucial alongside the actual content being conveyed.

Memes, combining persuasive discourse on social media platforms, prove particularly effective. They spread ideas or emotions online and are a popular tool in misinformation campaigns, using various rhetorical and psychological techniques to influence users. Memes' visual components either reinforce or convey persuasive tactics, thus playing a significant role in shaping public opinion and attitudes. To address these challenges, SemEval-2024 introduced a shared task focusing on detecting persuasion techniques from multilingual memes (Dimitrov et al., 2024). This task defines a hierarchy directed acyclic graph (HDAG) to represent a meme's persuasive techniques and highlights the challenges and importance of understanding the nuances of digital persuasion.

This study proposes exploring the effectiveness of multi-dimensional hierarchical classification (MDHC) strategies in identifying persuasive techniques in memes, based on previous research and the successful application of MDHC strategies in real-world HDAGs. The results from a competition show our approach's effectiveness, ranking first in a specific subtask and competitively across others, demonstrating the potential of MDHC strategies in analyzing persuasive discourse.

2 Background

In this study, we explore the application of Hierarchical Multi-label Classification (HMC) in the context of persuasive techniques, by structuring them within a hierarchical multi-label framework. This approach allows for the simultaneous handling of both textual and visual data through multimodal modeling.

Recent research (Montenegro et al., 2023) has demonstrated the efficacy of the MDHC approach for HMC, noting its simplicity and ease of implementation. HMC, an advancement of Multi-label Classification (MC), is designed to predict multiple labels that are organized hierarchically from general to specific categories. The incorporation of hierarchical knowledge is found to significantly improve the performance of classifiers.

The MC, which is applicable in a wide range of areas, involves the challenge of predicting multiple interrelated category variables. As highlighted by Alfaro et al. (Alfaro et al., 2023) and Bielza and Larrañaga (bie, 2011), the complexity of MC compared to single-dimensional problems is primarily

[1]https://huggingface.co/sdadas/xlm-roberta-large-twitter

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1868–1875

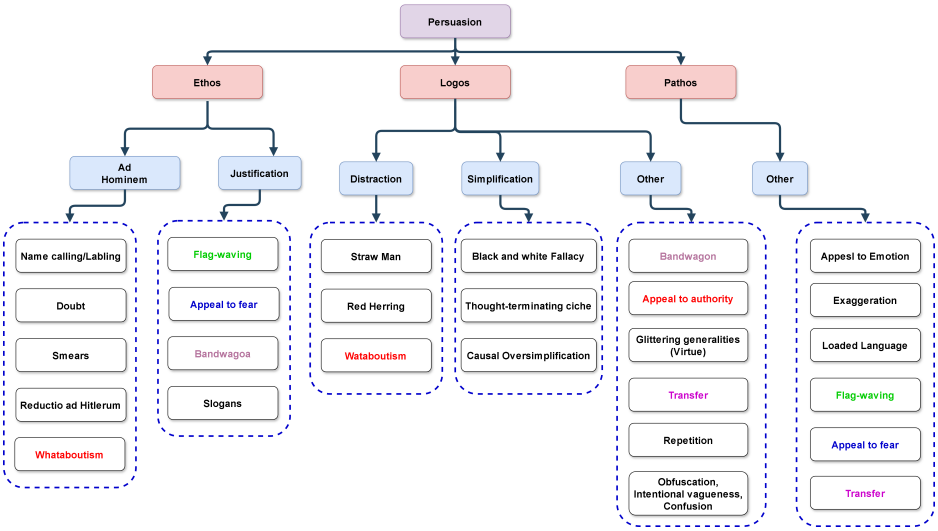


Figure 1: Hierarchy Multi-label Classification(HMC) with Persuasive Techniques

due to the vast combinations of class labels and the scarcity of relevant data.

We transform persuasive techniques into the HMC framework, this approach transforms persuasive technique graphs for application in specific subtasks. Given the necessity to analyze both textual and visual data for accurately identifying persuasive techniques, multimodal models become essential.

CLIP (Contrastive Language-Image Pre-Training) proposed by OpenAI (Radford et al., 2021), stands out for its independent text and image encoding capabilities, offering flexibility for various subtask types. In a study, (Kumar and Nandakumar, 2022) have suggested a range of techniques that combine textual and visual embedding vectors, leading to the effective detection of hateful memes. They also conducted various fusion experiments by switching different text encoders. Therefore, we refer to the authors' approach to combine the embedding vectors of the CLIP and the multilingual model with the aim of better adapting to Subtask 2ab, which involves tasks belonging to the multilingual domain and including datasets in three non-English languages (Bulgarian, North Macedonian, Arabic) in the test set.

3 Exploratory Data Analysis for Datasets

The dataset used in this study contains about 15,000 memes in English and other languages.

We have examined the label distribution for each task, and it is apparent that the data sets for subtask 1 and 2a are imbalanced, which differing numbers

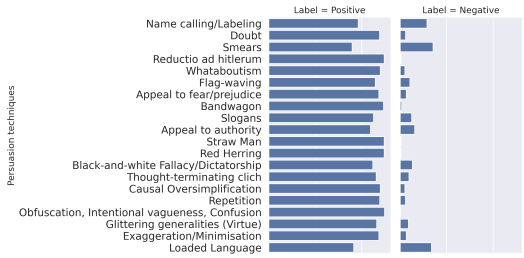


Figure 2: Data distribution of Persuasive Techniques on Subtask 1 Train Set

of Memes's persuasive techniques available for positive and negative are shown in Figure 2 and Figure 6. Further scrutiny, as delineated in Attachment Figure 5, it's evident that the datasets for Subtask 1 and 2a have a highly imbalanced distribution of data across 22 persuasion techniques, with Subtask 2a, in particular, showing a significant imbalance between positive and negative samples. We will describe how to address these imbalances in subsequent sections.

3.1 Transform the Structure of the Persuasive Techniques

The official release includes HDAG comprising 22 types of persuasive techniques. We have transformed this hierarchy into HMC. As shown in Figure 1, our reconstructed HMC has three levels: it includes 1 root node, the first layer has 3 child nodes, the second layer has 5 child nodes, and the bottom layer consists of 22 leaf nodes :

- **Root**: This describes whether a Meme image possesses any persuasive techniques.

- **First Layer Nodes**: There are 3 child nodes at this layer: Ethos, Logos, and Pathos. These nodes categorize the 22 types of persuasive techniques into 3 distinct classes of persuasive strategies.

- **Second Layer Nodes**: This layer includes 5 child nodes: Ad Hominem, Justification, Distraction, Simplification, and Other. We simplify the official hierarchy of persuasive techniques by using the "Other" node to encompass the Distraction and Simplification nodes, as they are redundant in the MDHC strategy.

- **Leaf Nodes**: There are 22 nodes at this level, corresponding to the 22 types of persuasive techniques that are the focus of this task. When a persuasive technique belongs to multiple categories in the first layer, it is represented by the same color.

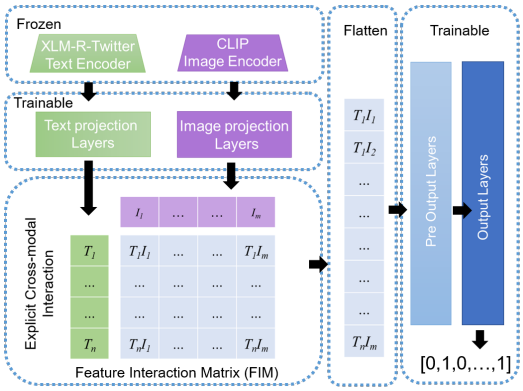


Figure 3: The Workflow for Multiclass Classification Task on the Multimodal Model

4 System Overview

In our research, we conducted an in-depth comparison of two MDHC strategies: Stacking+GC and Stacking+LCL, utilizing the same dataset for model training. The comparative analysis revealed that Stacking+GC demonstrated superior performance over Stacking+LCL. This superiority is attributed to its more effective handling of errors during the merging process of hierarchical levels, thereby enhancing the overall classification accuracy within the hierarchical structure of the data.

XLM-RoBERTA-large-twitter[1] For this system task, which is multilingual and specifically focused on the social media domain of Memes, we fine-tuned the domain-specific language model XLM-RoBERTA-large-twitter. This model was adjusted based on a corpus of over 156 million tweets in ten languages.

CLIP uses two distinct architectures as the backbone for encoding visual and textual datasets: image encoder, which represents the neural network architecture responsible for encoding images (e.g., ResNet or Vision Transformer), and text encoder, which represents the neural network architecture responsible for encoding textual information (e.g., BERT or Text Transformer). This structure is flexibly adapted to the subtasks of this project.

4.1 Detailed Description of the MDHC Strategy

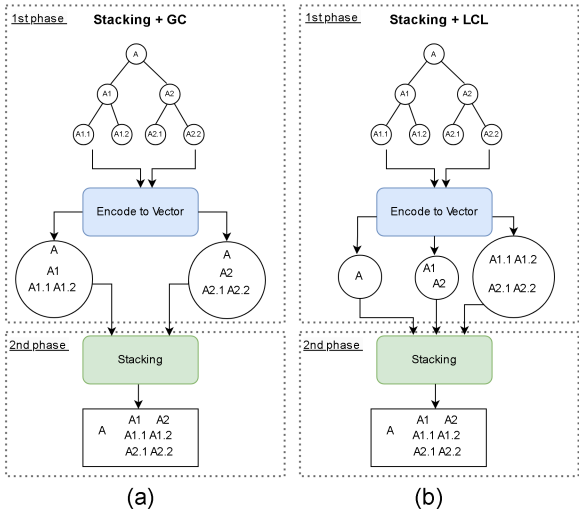


Figure 4: Illustration of the two DMC strategies. The feature vector is used on the 1^{st} phase as input, while the 2^{st} phase uses the outputs of the 1^{st} phase. it is to solve all the tasks associated with the internal nodes drawn inside the circle.

MDHC Strategies Incorporating the MDHC paradigm proposed by (Montenegro et al., 2023), our strategy selects an HMC classification approach suitable for this task, integrating both MC and HC strategies.

In MC strategy, we explore two solution algorithms: **Local Classifier per Level (LCL)**, which creates a model for each level of the hierarchy, and **Global Classifier (GC)**, a model that learns and predicts across the entire class hierarchy.

For HC strategy, we adopt *Stacking*, as introduced by (Wolpert, 1992), leveraging predictions from other labels to refine initial predictions, us-

ing average confidence scores to identify specific persuasive techniques with a threshold of 0.5 for determination.

To compare MDHC classification strategies, we integrate two MC strategies with an HC strategy, resulting in two distinct approaches.

Stacking + GC : this strategy applies the GC strategy to each dimension in the 1^{st} phase, utilizing feature vectors as inputs. The 2^{nd} phase employs the Stacking strategy, concatenating the probability vectors of the 1^{st} phase classifier's predictions for output. This strategy is shown in a Figure 4 (a)

Stacking + LCL : this strategy applies the LCL strategy to each dimension in the 1^{st} phase. In the 2^{nd} phase, it also adopts the Stacking strategy, using concatenated probability vectors from the first stage's predictions, with each circle representing a classifier addressing the classification problem of listed parent nodes. This strategy is shown in a Figure 4 (b)

4.2 Internal Negative Data Augmentation

In typical datasets, there is usually a similar ratio of positive to negative samples, even though they may not be evenly distributed. However, in the data distribution for Subtask 2b, there are only 2 samples that do not contain persuasive elements. Generally, one could use the PTC[2] dataset to augment this. However, the PTC dataset consists of news sentences with 18 types of persuasive techniques, and in hierarchical multi-label classification, the multiple labels have complex relationships.

When attempting to augment data by adding more examples for underrepresented labels, one must navigate the complex interplay between these labels carefully. Simply increasing the number of samples for a specific label can inadvertently exacerbate the imbalance for others. For example, if we augment the dataset with more instances of the 'Whataboutism' technique without considering its relationship with other techniques like 'Loaded Language' or 'Flag-waving,' we might skew the dataset further, making it even more challenging to train a balanced and accurate classifier.

To address this issue, we used two MDHC strategies. Firstly, we divided the entire hierarchy of persuasive techniques into six tasks based on the second-layer parent nodes: Ad Hominem, Justification, Distraction, Simplification, and Other (which was further split into two tasks). Each task independently applies the MC strategy, considering its specific persuasive techniques as positive examples and the others as negative examples, which approach we called **Internal Negative Data Augmentation(INDA)**, not only offers effective negative examples but also ensures consistency in labeling across various datasets. Ultimately, these six MC tasks determine the classification of the top-level parent node (indicating the presence of persuasive techniques) through a voting mechanism.

However, the INDA, while addressing the issue of imbalanced label distribution among samples without persuasive techniques, introduces the Long Tail Distribution problem. Long Tail Distribution is a probability distribution model characterized by lower probability density in its tail. In many cases, the distribution's right tail is considered more significant, but the left tail has a higher probability density. As shown in Attachment Figures 7, 8 and 9, the MDHC classification strategy we employed results in the positive distribution of a particular persuasive technique type being concentrated in the tail, forming a left-skewed long tail distribution.

The long-tail distribution presents two main challenges: Label co-occurrence and Dominance of negative labels:

Label co-occurrence : texts are often associated with multiple persuasive techniques simultaneously, making it difficult to accurately sample individual categories.

Dominance of negative labels : a text may only be associated with a small subset of persuasive techniques, resulting in the majority of labels being negative. However, Binary Cross-Entropy (BCE) treats positive and negative associated categories equally, leading to a shift in the boundary of negative associations.

To address the issues of Label co-occurrence and Dominance of negative labels, we introduce the Distribution-Balanced Loss (DBL) proposed by (Wu et al., 2020), the loss function employs re-balanced weighting and negative-tolerant regularization to mitigate the challenges posed by Label co-occurrence and Dominance of negative labels.

Therefore, without relying on external data augmentation, we utilize the MDHC strategy combined with the DBL loss function to address the sample imbalance issues in Subtask 1 and 2a. Additionally, we also introduce the DBL loss function to tackle the long-tail distribution problem.

[2]https://propaganda.math.unipd.it/ptc/

4.3 Meme with Multimodal Learning

In Subtask 2a and 2b, we employ the CLIP multimodal model, which includes an image encoder and a text encoder. The image encoder is responsible for encoding images, while the text encoder handles encoding textual information. At this stage, we utilize the XLM-RoBERTA-twitter fine-tuned on Subtask 1 as the text encoder because this model already possesses a certain understanding of meme text. For the image encoder, we use the pre-trained CLIP image encoder provided by the official source. Through Feature-wise Linear Modulation (FIM), these two encoders encode to obtain a representation embedding vector containing both image and text, enabling the model to effectively comprehend memes reliant on the relationship between text and image, such as persuasive techniques like Transfer and Appeal to (strong) Emotions.

Specifically, as depicted in Figure 3, Subtask 2ab involves encoding Meme images using the image encoder (I) and Meme text using the text encoder (T). Thus, we obtain sets of image encoding vectors $I_1...I_N$ and text encoding vectors $T_1...T_N$. We compute the FIM by multiplying these two vectors to obtain a new set of feature vectors. Subsequently, we attach a linear layer for classification at the end of the model to output the correct classifications. Specifically, in Subtask 2a, the learning objective is multi-label (MC) persuasion techniques classification, while in Subtask 2b, the learning objective is binary classification to identify whether it contains persuasive techniques or not.

5 Experiment and Evaluate

In our experiments, we employed three MDHC classification strategies on subtasks. All tasks utilize the HierarchyF_1 evaluation metric, which is the unified evaluation metric provided by the official source. In the experiment setting, refer to Appendix A for details of the relevant parameters.

In **Subtask 1**, we compared the performance of two language models, XLM-RoBERTA and XLM-RoBERTA-Twitter, across Subtask 1 and 2a, to assess the impact of domain-specific pre-training. We explored three classification strategies: GC, Stacking + GC, and Stacking + LCL, to identify the most effective approach for Subtask 1. In **Subtask 2a**, we evaluated two multimodal models by combining CLIP with XLM-RoBERTA-Twitter from Subtask 1. The goal was to improve the comprehension of Meme's persuasive techniques by utilizing a fine-tuned text encoder. Additionally, we employed GC, Stacking + GC, and Stacking + LCL strategies to determine which one is more effective for subtask 2a. Our primary focus was on improving classification performance in a multimodal setting. In **Subtask 2b**, we explored multimodal model and classification strategy using the CLIP + XLM-RoBERTA-Twitter combination from subtask 1, applied to a binary classification framework. Utilizing a balanced dataset of hate Memes collected by Meta AI, as referenced in the Harmful Memes Dataset (Kiela et al., 2020), the goal was to train the model on balanced samples to prevent bias effectively.

Table 1: Performance for Subtask 1 in Validation Set

Models	Strategy	Metrics		
		H-F1	H-Prec	H-Rec
Baseline	Official	0.3651	0.4573	0.3038
XLM-R Large	GC	0.5594	0.4635	0.6335
	Stacking + LCL	0.5995	0.5244	0.6909
	Stack + GC	0.6262	0.5907	0.6646
XLM-R-Large Twitter	GC	0.5580	0.6367	0.6503
	Stacking + LCL	0.6310	0.6062	0.6764
	Stacking + GC	**0.6689**	**0.6843**	**0.7451**

Table 2: Performance for Subtask 2a in Validation Set

Models	Strategy	Metrics		
		H-F1	H-Prec	H-Rec
Baseline	Official	0.4589	0.6820	0.3457
CLIP + XLM-R Large	GC	0.5214	0.6320	0.5775
	Stacking + LCL	0.6265	0.6343	0.5947
	Stack + GC	0.6675	0.7598	0.6107
CLIP + XLM-R-Large Twitter	GC	0.5581	0.6369	0.6103
	Stacking + LCL	0.6567	0.6459	0.6178
	Stacking + GC	**0.7134**	**0.7652**	**0.6418**

Table 3: Performance for Subtask 2b in Validation Set

Models	Strategy	Metrics	
		F1 macro	F1 micro
Baseline	Official	0.2500	0.3333
CLIP + XLM-R-Large	Binary Classification	0.7618	0.7947
CLIP + XLM-R-Large Twitter	Binary Classification	**0.8023**	**0.8216**

Results Our study's evaluation of the official validation set showcases the impactful of domain knowledge in enhancing model performance for persuasive technique identification across various subtasks. **Subtask 1**: As shown in Table 1, the model fine-tuned with domain knowledge performs significantly better in identifying persuasive techniques. Among the three classification strategies, Stack + GC demonstrates superior performance compared to the other two strategies. **Subtask 2a**:

The evaluation results for this task, as depicted in Table 2, align with Subtask 1. The model fine-tuned with domain knowledge outperforms others in identifying persuasive techniques. Among the three classification strategies, Stack + GC exhibits superior performance. **Subtask 2b**: In Table 3, this task involves binary classification. We utilized a multimodal model for binary classification and achieved competitive scores through external data augmentation methods.

Table 4: Performance for Subtask 1 in Test Set

Languages	Method	Metries		
		H-F1	H-Prec	H-Rec
English	Baseline	0.36865	0.47711	0.30036
	Ours	**0.66271**	**0.60990**	**0.72552**
Bulgarian	Baseline	0.28377	0.31881	0.25567
	Ours	**0.51744**	**0.53578**	**0.50031**
North Macedonian	Baseline	0.30692	0.31403	0.30012
	Ours	**0.46165**	**0.54622**	**0.39975**
Arabic	Baseline	0.35897	0.35000	0.36842
	Ours	**0.47500**	**0.42817**	**0.53333**

Table 5: Performance for Subtask 2a in Test Set

Languages	Method	Metries		
		H-F1	H-Prec	H-Rec
English	Baseline	0.44706	0.68778	0.33116
	Ours	**0.70677**	**0.78164**	**0.64498**
Bulgarian	Baseline	0.50000	**0.80428**	0.36276
	Ours	**0.54864**	0.70691	**0.44828**
North Macedonian	Baseline	**0.55525**	**0.90219**	**0.40103**
	Ours	0.48707	0.70575	0.37185
Arabic	Baseline	**0.48649**	**0.65000**	0.38870
	Ours	0.48323	0.59466	**0.40698**

Table 6: Performance for Subtask 2b in Test Set

Languages	Method	F1 macro	F1 micro
English	Baseline	0.25000	0.33333
	Ours	**0.78803**	**0.82167**
Bulgarian	Baseline	0.16667	0.20000
	Ours	**0.64706**	**0.82000**
North Macedonian	Baseline	0.09091	0.10000
	Ours	**0.52000**	**0.79000**
Arabic	Baseline	0.22705	0.29375
	Ours	**0.58518**	**0.59375**

In the test set of the competition, we propose a method that has demonstrated competitive performance on the official competition leaderboard. As shown in Table 4, our method outperforms all official baselines in Subtask 1, which involves the identification of meme persuasion techniques in four different languages at the text level. Our method ranks 4th on average across four languages (English, Bulgarian, North Macedonian, and Arabic), with rankings of 6th in English, 3rd in Bulgarian, 4th in North Macedonian, and 2nd in Arabic. These results indicating our method is competitive.

In the multimodal task, as shown in Table 5, We observe the performance of our method in Subtask 2a, where it demonstrates competitiveness in English memes. We attribute this to two main reasons. Firstly, our method only undergoes text-level domain pretraining on the English memes provided in the training dataset. As a result, it lacks the necessary representation capabilities for low-resource languages, such as North Macedonian. based on the above, our method does not improve cross-linguistic abilities; instead, it relies on the language representation capabilities of the multilingual model. Therefore, this makes it challenging to identify cross-lingual fine-grained persuasion techniques. Secondly, the inherent cultural differences in various languages may lead to discrepancies in the same set of memes presented in different languages, indicating a potential issue of cultural divergence.

Finally, in Subtask 2b, as shown in Table 6, although our proposed method falls short in the fine-grained cross-linguistic meme persuasion techniques identification, it remains competitive in tasks involving the identification of whether a meme, combining text and image, contains one of the persuasion techniques. Our method ranks 6th on average across four languages (English, Bulgarian, North Macedonian, and Arabic), with rankings of 7th in English, 5th in Bulgarian, 6th in North Macedonian, and 5th in Arabic. This demonstrates the competitive edge of our method in a multimodal context.

6 Conclusion

We conducted a detailed analysis of the HDAG containing persuasive techniques, transforming it into an HMC task. We also explored two MDHC strategies and highlighted the importance of addressing the long-tail distribution issue, proposing the use of the DBL loss function to mitigate this issue in HMC tasks. Regarding the models, we recommend utilizing domain-specific pre-training to detect memes containing persuasive elements and the effectiveness of domain-specific training was demonstrated across various experiments. Finally, we achieve competitive results in the competition.

References

2011. Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, 52(6):705–727.

Juan C Alfaro, Juan A Aledo, and José A Gámez. 2023. Multi-dimensional bayesian network classifiers for partial label ranking. *International Journal of Approximate Reasoning*, page 108950.

Dimitar Dimitrov, Giovanni Da San Martino, Preslav Nakov, Firoj Alam, Maram Hasanain, Abul Hasnat, and Fabrizio Silvestri. 2024. Semeval-2024 task 4:multilingual detection of persuasion techniques in memes. *In Proceedings of the 18th International Workshop on Semantic Evaluation*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624.

Gokul Karthik Kumar and Karthik Nandakumar. 2022. Hate-CLIPper: Multimodal hateful meme classification based on cross-modal interaction of CLIP features. In *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 171–183, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Robin Tolmach Lakoff. 1982. Persuasive discourse and ordinary conversation, with examples from advertising. *Analyzing discourse: Text and talk*, pages 25–42.

C Montenegro, R Santana, and JA Lozano. 2023. Introducing multi-dimensional hierarchical classification: Characterization, solving strategies and performance measures. *Neurocomputing*, 533:141–160.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. 2020. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 162–178. Springer.

A Implementation Details

During training, we use AdamW as the optimizer and an initial learning rate of 2e-5 for XLM-Roberta-twitter and 1e-4 for CLIP models. with a batch size of 32 and text max length set to 128 on subtask 1 and a batch size of 16, image size set to 224, and text max length set to 128 on subtask 2a and 2b. with all subtasks, the maximum number of epochs is set to 50. All experiments are conducted using two NVIDIA TITAN RTX GPUs

B Data Distribution Details

B.1 Data Distribution of the Persuasion Techniques

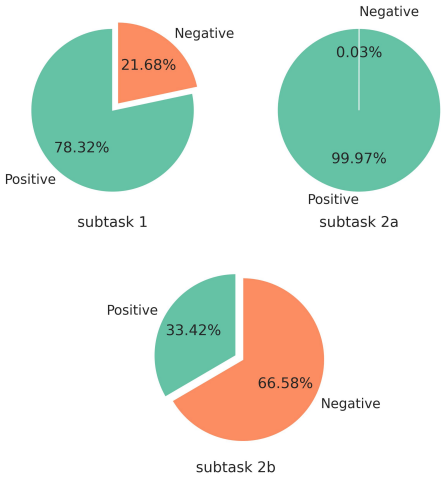


Figure 5: Ratio of the Persuasion Techniques on Subtask 1,2a and 2b

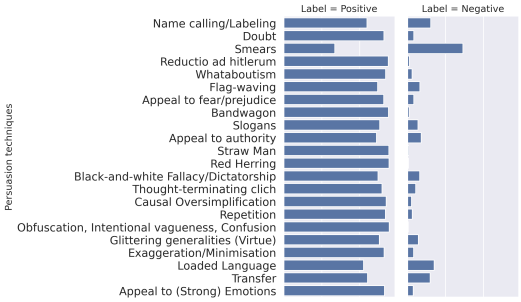


Figure 6: Data Distribution of Persuasion Techniques on Subtask 2a Train set

B.2 Data Distribution of MC for Persuasion Techniques in the MDHC strategy

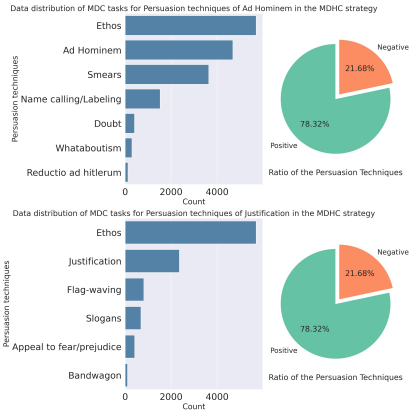


Figure 7: Data Distribution of MC for Persuasion Techniques of Ethos in the MDHC Strategy on Subtask 2a

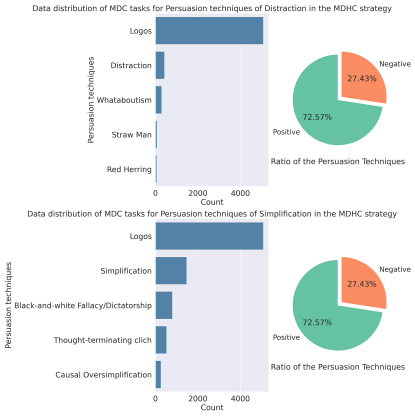


Figure 8: Data Distribution of MC for Persuasion Techniques of Logos in the MDHC Strategy on Subtask 2a

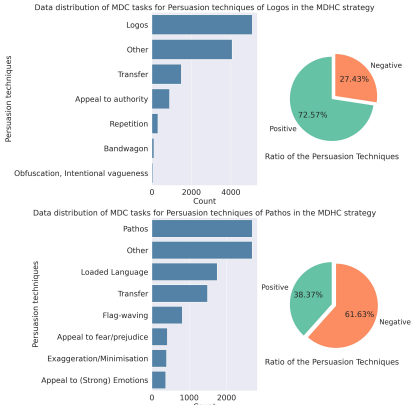


Figure 9: Data Distribution of MC for Persuasion Techniques of Pathos and Logos in the MDHC Strategy on Subtask 2a

Mothman at SemEval-2024 Task 9: An Iterative System for Chain-of-Thought Prompt Optimization

Alvin Po-Chun Chen and **Ray Groshan** and **Sean von Bayern**
University of Colorado Boulder
{alvin.chen, ray.groshan, sean.vonbayern}@colorado.edu

Abstract

Extensive research exists on the performance of large language models on logic-based tasks, whereas relatively little has been done on their ability to generate creative solutions on lateral thinking tasks. The BRAINTEASER shared task tests lateral thinking and uses adversarial datasets to prevent memorization, resulting in poor performance for out-of-the-box models. We propose a system for iterative, chain-of-thought prompt engineering which optimizes prompts using human evaluation. Using this shared task, we demonstrate our system's ability to significantly improve model performance by optimizing prompts and evaluate the input dataset.[1]

1 Introduction

The ability for language models to reason or possess common sense knowledge has become a controversial topic with far-reaching implications (Bender and Koller, 2020). Large language models (LLMs) show remarkable results on *vertical thinking* tasks that require sequential logical inference (Liu et al., 2019) but there have been relatively few studies done on *lateral thinking* puzzles—tasks that require more creative, "outside the box" problem-solving processes. As larger LLMs with the ability to memorize large corpora (Hartmann et al., 2023) are developed, lateral thinking tasks become an increasingly important benchmark for analyzing and evaluating their reasoning capacities. The BRAINTEASER shared task (Jiang et al., 2023)(Jiang et al., 2024) is designed to elicit and evaluate lateral thinking through two English-language subtasks, using sentence puzzles and word puzzles respectively.

In this paper, we propose a novel method for optimizing chain-of-thought (CoT) prompting (Wei et al., 2023) on the `GPT-4` model which we use to tackle the sentence puzzle subtask. Our system iteratively optimizes CoT prompting by systematically evaluating input data and model output using human performance as a benchmark. We identify question types that are difficult for humans, informing the next iteration of prompt engineering. Not only does this process optimize CoT prompting for a specific task, our system also provides insights for improving future data collection and synthesis.

Our main contribution is the novel approach for identifying reasoning challenges to optimize prompting. For the sentence-based task, we develop a prompt engineering method which requires the model to reason over all answer choices and provide explanations for both correct and incorrect choices. In doing so, the model is more likely to refute choices that are semantically related to the question but logically incorrect. Our methodology significantly improves performance for adversarial datasets and achieves more consistent results, which suggests that the model relies less on memorization when using these CoT prompts.

As part of our evaluation of the data, we also identify several questions in the adversarial datasets that are difficult to solve due to having multiple logical options or are unanswerable with the provided premises. By combining model reasoning with human evaluation, we can quickly identify and evaluate problematic questions. This process can further explain model performance and provide guidance for future data collection/generation.

2 Background

Question Answering (QA) is a well-established task in natural language processing with broad applications both in academia and in industry (Hirschman and Gaizauskas, 2001). Recent work such as CommonSenseQA (CSQA) (Talmor et al., 2018) and StrategyQA (Geva et al., 2021) focus on reasoning questions that require logical inference

[1]Our code can be found at https://github.com/alvin-pc-chen/semeval_brainteaser.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1876–1888

in the form of vertical thinking. BRAINTEASER questions instead require lateral thinking to answer, much like questions in the traditional "brainteaser" style(Jiang et al., 2023)(Jiang et al., 2024):

Base: Samuel was out for a walk when it started to rain. He did not have an umbrella and he wasn't wearing a hat. His clothes were soaked, yet not a single hair on his head got wet. How could this happen?

1. His hair is dyed.
2. **This man is bald.**
3. This man walk upside down to avoid rain.
4. None of above.

SR: Rain began to fall as Samuel was taking a stroll. He wasn't wearing a hat, and he didn't have an umbrella. Even though his clothes were completely drenched, not a single hair on his head was moist. How is this even possible?

1. This man walk upside down to avoid rain.
2. His hair is dyed.
3. **This man is bald.**
4. None of above.

CR: Tom is a clean freak but he never dries his hair after a shower. How is this possible?

1. His hair is dyed.
2. He tries to stand upside down during shower to avoid rain.
3. **This man is bald.**
4. None of above.

The data for this subtask is drawn from online English-language riddles and brainteasers, with incorrect choices created by handpicking entailments generated by COMET (Bosselut et al., 2019) using incorrect premises. Each question has three unique answers, as well as a shared fourth option, "None of above". To counter memorization from LLMs trained on web crawls, the task authors generated two synthetic datasets using *semantic reconstruction* (SR) and *context reconstruction* (CR). The SR dataset rephrases the original question without changing the answer or premises while the CR dataset changes the situational context without changing the misleading premise. The SR dataset was generated with an open-source rephrasing tool while the CR dataset was generated using GPT-4; both sets were manually refined by human annotators. In total, 208 question/answer pairs were sampled for the base set resulting in 624 questions after SR and CR augmentation. The training set was split with 81.25% of the data with the same base/SR/CR questions kept together in the split.

Although the task is designed to elicit lateral thinking, we consider an alternative understanding of the task by thinking of the questions as *noisy*. Questions are loaded with irrelevant, contradictory, or misleading information to distract the respondent. Since transformers generally learn meaning by scoring tokens across the sentence or sentence pair, they are biased against long-tail knowledge (Li et al., 2023), which is knowledge that occurs infrequently in the training set.

Brainteasers, by their nature, rely on the unconventional interpretation of the question to stump the respondent. This same property can trick the model into selecting a semantically similar answer choice that is logically incorrect. Chain-of-Thought (CoT) prompt engineering (Wei et al., 2023) is a recent method that has been shown to not only improve outcomes on similar problems, but also to provide an interpretable window for human review. CoT prompts provide example questions with related reasoning to the model, which induces the model to provide reasoning for a given answer in the output. We utilize both of these properties to introduce an iterative method that optimizes CoT prompting for a given task.

3 System Overview

We propose an iterative system for optimizing the CoT prompt engineering process:

1. Randomly sample the training data and naively engineer CoT prompts.
2. Identify distinct categories in output reasoning to partition training data.
3. Perform independent human evaluation to isolate specific challenges in each category.
4. Use findings to inform the development of new CoT prompts.
5. Optionally, identify gaps in the data for future data collection/synthesis.

Each step of our process is iterative, although independent human evaluation should only be performed when novel problem categories are identified in the model reasoning. Once the human benchmark has been incorporated into the prompt engineering process, future iterations mainly rely on evaluating model outputs for gaps in logic. All evaluation steps can provide powerful insights into the dataset itself to inform future dataset creation,

and is particularly useful for real-world applications where data selection is an open-ended problem. By identifying gaps and problems in the data, more representative data can be collected to improve model performance on the given application. This can be thought of as a backpropagating human and model outputs back to the prompt engineering and data curation steps.

3.1 Naive Chain-of-Thought Prompting

In the first step, we randomly sample the training data to generate naive CoT prompts to use on the test set. This step eliminates a large portion of the dataset that naive CoT prompting already solves while also providing outputs with interpretable windows for identifying problem questions and corresponding failure in logic. For example, the topics (mathematics, physics, law, etc.) identified by the task authors in the training data were found to have minimal impact on model accuracy. Instead, we found that the construction type impacted model performance above all. By focusing on the model outputs, we determine that the type of reconstruction (base, SR, and CR) have the highest impact on model performance and require human analysis.

For the first round of CoT prompting, we randomly select 8 samples from the training set and generate the Naive CoT-Base prompt set based on the logical premises of the questions (all naive prompts in Appendices A):

```
Naive CoT Example Prompt:
Question: A horse is tied to a five-meter rope in front of an old saloon. Ten meters behind the horse is a bale of hay. Without breaking his rope, the horse is able to eat the hay whenever he chooses. How is this possible?
Choices:
0 = The rope stretches proportionally, providing the extra length needed for the horse to reach the hay ten meters away.;
1 = The rope is not tied to anything else.;
2 = The walls of the saloon retract or collapse inwards, creating more space for the horse to reach the hay.;
3 = None of above.;
Response: That the rope is not tied to anything else is the simplest choice. The horse can reach the hay whenever he chooses. The answer is 1
```

3.2 Human Evaluation Step

Based on model performance using the naive CoT prompts, we separate the test set along base, SR, and CR lines for human testing. When prompting `GPT-4`, each question is independently shown to the model with no retention in between. Humans are not under similar constraints and can easily identify reconstructions, especially the SR set which share the same answer choices with the base set. For accurate comparison, we select different participants to answer each dataset.

We selected 3 participants for each dataset in order to collect robust results while still maintaining consensus. All participants are graduate students with native proficiency in English, and surveys were completed using Google Forms with the same instructions, randomized question order, and constant answer choice order. An additional option "Unsure" was provided to uncover difficult questions which was counted as "None of above" for testing purposes.

We analyze accuracy along four metrics: mean, minimum score, maximum score, and consensus score. The minimum score is counted only when all participants answer correctly whereas the maximum score is counted when any participant answers correctly. The consensus score uses the answer selected by 2/3 participants; 4 questions had no consensus and were marked as incorrect. Through this process, we identified common errors for humans and models in each dataset which informed the second iteration of CoT prompting.

3.3 Iterated Chain-of-Thought Prompting

Besides the tricky CR questions uncovered in the human evaluation step, we also identified that our naive sample was overly weighted on base questions, which potentially serves to reinforce model memorization. To address both issues, we develop the CoT-Mix set, a new set of 8 prompts weighted towards SR and CR questions and tailored towards disproving incorrect answer choices. Since our human benchmark performed particularly poorly on CR questions, we also separately created CoT prompts comprised entirely of base, SR, and CR sets for further comparison. All prompt sets can be found in the Appendices B.

```
Iterated Chain-of-Thought Prompt:
Question: A horse is tied to a five-meter rope in front of an old saloon. Ten meters behind the horse is a bale of hay. Without breaking his rope, the horse is able to eat the hay whenever he chooses. How is this possible
Choices:
0 = The rope stretches proportionally, providing
```

System	Instance Based			Group Based		Overall
	Base	SR	CR	Base&SR	Adversarial	
abdelhak	100	100	95.0	100	95.0	98.3
Human (Jiang et al., 2023)	90.7	90.7	94.4	90.7	88.9	92.0
Human Consensus (Ours)	90.0	90.0	67.5	80.0	55.0	82.5
GPT-4 Zero Shot	87.5	72.5	70.0	72.5	60.0	76.7
GPT-4 Multi Shot Base	92.5	**90.0**	80.0	**87.5**	70.0	87.5
GPT-4 Multi Shot Mix	**95.0**	**90.0**	**85.0**	**87.5**	**80.0**	**90.0**
GPT-4 Naive CoT-Base	**95.0**	**87.5**	75.0	85.0	65.0	85.8
GPT-4 Naive CoT-Mix	92.5	**87.5**	**82.5**	**87.5**	**75.0**	**87.5**
GPT-4 New CoT-Base	**97.5**	85.0	80.0	85.0	70.0	87.5
GPT-4 New CoT-SR	90.0	90.0	75.0	85.0	67.5	85.0
GPT-4 New CoT-CR	92.5	90.0	77.5	87.5	67.5	86.7
GPT-4 New CoT-Mix	95.0	**92.5**	**82.5**	**92.5**	**77.5**	**90.0**

Table 1: Accuracy of each model; **Base**, **SR**, and **CR** are scored on individual datasets, **Base&SR** only counts if both the base and SR versions are correct for a given question, **Adversarial** only counts if all three versions are correct for a given question, and **Overall** counts base, SR, and CR separately.

```
the extra length needed for the horse to reach the
hay ten meters away.;
1 = The rope is not tied to anything else.;
2 = The walls of the saloon retract or collapse
inwards, creating more space for the horse to reach
the hay.;
3 = None of above.;
Response: Rope generally cannot stretch, and if it
could stretch the length would be variable. If the
walls collapse, the horse would be further from
the hay. The rope not being tied to anything else
is the simplest answer. The answer is 1
```

4 Experimental Setup

All experiments were performed using GPT-4 via the OpenAI API[2] with the same system prefix. Each question was called separately with the respective prefix. Initial experiments limited token count to 1 in order to force the model to output integer labels but the restriction was removed due to poor performance. Output text was logged and labels were extracted deterministically; labels that could not be extracted this way were reviewed and manually entered. Results shown in this paper were from API calls between 2024/01/15-2024/02/17; since OpenAI models are updated regularly replication results may differ.

The input data was first preprocessed by removing extra spaces, lines, punctuation, and spelling and grammatical errors. Due to the variety of sentence-based problems in the data, some answer choices were multiple sentences long while others were single words; this discrepancy could potentially affect model performance. Since semicolons do not occur in the data at all, they were selected as separators between answer choices for GPT-4 prompting to mitigate this issue. System prompts can be found in Appendix C.

[2] https://platform.openai.com/

5 Results

In this section, we compare results across four categories: human performance, zero/multi-shot performance, naive CoT prompt performance, and the iterated CoT prompt performance. For completion, we also provide the top competition result (abdelhak). Our official submission for the shared task leaderboard used the GPT-4 New CoT-Base prompts. With 31 participants, our results scored 2nd overall on the base data, 7th on SR, CR, and Base&SR accuracy, 9th on adversarial accuracy, and 9th overall.

5.1 Quantitative Benchmarks

As expected, the new CoT prompts show significant improvements along all metrics compared to their naive counterparts. Since the adversarial datasets are designed specifically to counteract model memorization, the gains made in the group-based metrics between the Naive and New CoT-Mix prompts in particular demonstrates the effectiveness of our system. The CoT-Base sets, on the other hand, likely still suffer from overweighting on the original questions crawled from online

sources. The fact that multi-shot prompting outperforms the Naive CoT prompts further supports this argument. Despite not receiving any guidance on logic, the model is still able to achieve strong results on the task. However, once we introduce the idea of disproving incorrect answers, the model is once again able to make gains in performance.

Interestingly, the SR and CR prompts did not show significant improvements compared to the base set and even performed worse on the CR questions. This could potentially be due to the fact that the model performs worse on these question types overall. Without base questions to provide a foundation, the model is unable to generate the most robust reasoning. While this finding provides grounds for further exploration, the results from the New CoT-Mix prompting shows consistent improvements across the board, suggesting that there are commonalities within each adversarial dataset that can be identified by the model.

5.2 Human Performance Evaluation

Dataset	Mean	Min.	Con.	Max.
Base	84.2	65.0	87.5	100.0
SR	85.8	70.0	90.0	97.5
CR	60.0	30.0	65.0	80.0

Table 2: Human participant accuracy for each dataset; **Min.** is only counted when all three participants answer correctly, **Con.** is counted when 2/3 agree on an answer, and **Max.** is counted when any participant scores correctly.

When evaluating human performance, we find that there were significantly higher rates of "Unsure" answers in the CR set which contributes to the lower overall score. Along with other observations, the greater rate of "Unsure" answers supports the idea that the CR questions are more difficult to reason over. However, there were no cases where all respondents selected "Unsure" for the same question, making this metric a weak indicator of problematic questions. One possible explanation suggests that the wording for the answers in the CR set are less well-formed than the other sets, leading to greater confusion among respondents.

For human performance, none of our CoT results were able to beat the task paper human benchmark on the CR dataset, although our iterated CoT results did surpass the human benchmark on SR. Our human results only drastically differ on the CR set with those found in the task paper, which could be influenced by several factors. A major factor is the fact that we use independent human annotators for each of the datasets, meaning that those working on the CR set have no reference to the original question. For questions with multiple valid answers or unclear logic, our annotators would not be able to reference the base and SR versions for clues. Out of 14 incorrectly answered CR questions, we identify 5 such questions.

Specifically, we discovered several questions in the CR set with multiple logical responses. In the example below, the correct answer is "two and a half hours" with the assumption that each stop is meant to take half an hour. Ignoring that the premise itself is nonsensical (the driver would have simply been told to stop for the full two and a half hours), nowhere is it stated that the driver takes half an hour for each break. As a result, both (2.) and (4.) are viable candidates depending on reasoning. The rest of the problem questions can be seen in Appendix D.

```
Example Question: A driver is told to make a stop
every half an hour for the engine to cool down,
for five times. How long do the stops take?
Choices:
1. Three hours.
2. Two and a half hours.
3. Two hours.
4. None of above.
```

6 Conclusions

In this paper, we demonstrate a novel system for optimizing chain-of-thought prompt engineering using human evaluation. While Naive CoT prompts performed similarly to multi-shot prompting without guidance on reasoning, later iterations were able to approach state-of-the-art performance. Significant improvement on the SR, CR, and group-based metrics were shown on the test data, supporting the adoption for this method of prompt engineering. This system further provides guidance on identifying key problem areas in the dataset, specifically with regards to the generation of context reconstruction questions. This form of evaluation serves to inform decisions for improving dataset quality. For future work, we plan to implement different techniques to create synthetic data and perform the same evaluation across an open-ended dataset.

References

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies.

Valentin Hartmann, Anshuman Suri, Vincent Bindschaedler, David Evans, Shruti Tople, and Robert West. 2023. Sok: Memorization in general-purpose large language models.

L. Hirschman and R. Gaizauskas. 2001. Natural language question answering: the view from here. *Natural Language Engineering*, 7(4):275–300.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Huihan Li, Yuting Ning, Zeyi Liao, Siyuan Wang, Xiang Lorraine Li, Ximing Lu, Faeze Brahman, Wenting Zhao, Yejin Choi, and Xiang Ren. 2023. In search of the long-tail: Systematic generation of long-tail knowledge via logical rule guided search.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *CoRR*, abs/1811.00937.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

A Naive CoT Prompts

A.1 Naive Base Prompts

1. Question: Samuel was out for a walk when it started to rain. He did not have an umbrella and he wasn't wearing a hat. His clothes were soaked, yet not a single hair on his head got wet. How could this happen?
   Choices:
   0 = His hair is dyed.;
   1 = This man is bald.;
   2 = This man walk upside down to avoid rain.;
   3 = None of above.;
   Response: Samuel got wet from the rain, but no hair on his head got wet. If Samuel dyed his hair it would still be wet. Samuel is bald, so his hair didn't get wet. The answer is 1

2. Question: A horse is tied to a five-meter rope in front of an old saloon. Ten meters behind the horse is a bale of hay. Without breaking his rope, the horse is able to eat the hay whenever he chooses. How is this possible?
   Choices:
   0 = The rope stretches proportionally, providing the extra length needed for the horse to reach the hay ten meters away.;
   1 = The rope is not tied to anything else.;
   2 = The walls of the saloon retract or collapse inwards, creating more space for the horse to reach the hay.;
   3 = None of above.;
   Response: That the rope is not tied to anything else is the simplest choice. The horse can reach the hay whenever he chooses. The answer is 1

3. Question: A woman who lives in new york legally married three men, she did not get divorce, get an enrollment, or legally seperate. How is this possible?
   Choices:
   0 = The woman is not a good person.;
   1 = His husband is not a good husband.;
   2 = She is a minister.;
   3 = None of above.;
   Response: In new york it is not legal to be married to more than one person at a time. If the woman did not get divorced or legally separated, she cannot be legally married to more than one person. Ministers perform marriages, they do not get married. The answer is 2

4. Question: Why is the value of 1968 pennies higher than 1967 pennies?
   Choices:
   0 = Old money weight more than new money.;
   1 = Old money is dirtier than new money.;
   2 = Because there is one more penny in 1968 pennies than in 1967 pennies.;
   3 = None of above.;
   Response: 1968 and 1967 refer to the number of pennies. The value of 1968 pennies is higher than 1967 pennies because there is one more penny in 1968 pennies than in 1967 pennies. The answer is 2

5. Question: Not a single parent objected when the teacher spanked every child in the class. How come?
   Choices:
   0 = The teacher had informed all the parents in advance about the unique disciplinary approach.;
   1 = The teacher had informed all the children in advance about the unique disciplinary approach.;
   2 = The teacher was in an orphanage school.;
   3 = None of above.;
   Response: Even if the teacher had informed all the parents or all the students in advance about the unique disciplinary approach, it is unlikely that nobody would object. If the teacher is in an orphange school, there would be no parents to object. The answer is 2

6. Question: Twenty-seven ducks are going to the pond. Five of them got lost, thirteen of them are staying home, and nine of them are at the pond. Where are the rest of them?
   Choices:
   0 = Home.;
   1 = The way to the pond.;
   2 = Pond.;
   3 = None of above.;
   Response: Twenty-seven ducks are going to the pond. Five of them are lost, 27 - 5 = 22. Thirteen of them are staying home, 22 - 13 = 9. Nine of them are at the pond, 9 - 9 = 0. There are no more ducks so the question is invalid. The answer is 3

7. Question: How many birth days does the average person have?
   Choices:
   0 = People may celebrate their birthdays annually, so it depends on their life span.;
   1 = They technically only have one birth day in their lifetime.;
   2 = It can be zero as some people are too busy to celebrating their birthdays.;
   3 = None of above.;
   Response: Everyone is born only once so technically the average person has one birthday in their lifetime. The answer is 1

8. Question: The more you take, the more you leave behind
   Choices:
   0 = Love.;
   1 = Footsteps.;
   2 = Money.;
   3 = None of above.;
   Response: Footsteps. The more you walk, the more footsteps you leave behind. The answer is 1

A.2 Naive Mixed Prompts

1. Question: A horse is tied to a five-meter rope in front of an old saloon. Ten meters behind the horse is a bale of hay. Without breaking his rope, the horse is able to eat the hay whenever he chooses. How is this possible?
   Choices:
   0 = The rope stretches proportionally, providing the extra length needed for the horse to reach the hay ten meters away.;
   1 = The rope is not tied to anything else.;
   2 = The walls of the saloon retract or collapse inwards, creating more space for the horse to reach the hay.;
   3 = None of above.;
   Response: Rope generally cannot stretch, and if it could stretch the length would be variable. If the walls collapse, the horse would be further from the hay. The rope not being tied to anything else is the simplest answer. The answer is 1

2. Question: A woman who lives in new york legally married three men, she did not get divorce, get an enrollment, or legally seperate. How is this possible?
   Choices:
   0 = The woman is not a good person.;
   1 = His husband is not a good husband.;
   2 = She is a minister.;
   3 = None of above.;
   Response: In new york it is not legal to be

married to more than one person at a time, even if her husband is bad. If the woman did not get divorced or legally separated, she cannot be legally married to more than one person. Ministers perform marriages, they do not get married. The answer is 2

3. Question: Brad began by entering the office tower's 22nd story through the filthy common window. He slid the window open and leapt through it after becoming depressed. Outside the building, there was a cliff-like drop to the ground. Amazingly, he came to rest entirely unharmed. How could he have survived the fall if there was nothing to slow or cushion his descent?
   Choices:
   0 = Brad was so sick and tired of window washing, he opened the window and jumped inside.;
   1 = The ground outside the building is wet.;
   2 = Consistent exercise has made him a very strong man.;
   3 = None of above.;
   Response: Even very strong men would not survive a fall from a 22-story building. The ground outside the building being wet would not cushion his fall. Brad was so sick and tired of window washing, he opened the window and jumped inside. The answer is 0

4. Question: Danny had just passed under an overpass in his semi when he abruptly came to a stop. Danny accidentally drove under the overpass that was only just tall enough for his truck because he wasn't paying enough attention. He was unable to move forward or backward in the semi due to how tightly it was wedged. When another tracker passed by, he was told how simple it would be to remove the semi from underneath the bridge. What did he recommend?
   Choices:
   0 = He told Danny to left the bridge.;
   1 = He told Danny to overturn the track.;
   2 = He told Danny to let some air out of his tires.;
   3 = None of above.;
   Response: The bridge would not be lifted. Overturning the truck would not help. Letting some air out of his tires would help. The answer is 2

5. Question: I excavate little caves and keep my gold and silver there. I also create gold crowns and silver bridges. They are the tiniest things you can imagine. Everyone will eventually require my assistance, but many people are reluctant to accept it. Why?
   Choices:
   0 = I am a dentist.;
   1 = I am a thief.;
   2 = I am a miner.;
   3 = None of above.;
   Response: Thieves and miners do not create gold crowns or bridges. Tiny crowns and bridges are dental work. People don't like going to the dentist. The answer is 0

6. Question: Each of the 30 participants in the masquerade had to wear a unique hat to distinguish themselves from one another. The host, however, only tallied 29 when he counted the number of hats to determine attendance. All 30 persons had signed their names on the spreadsheet, which confused him. He repeated the count. There are still 29. How is that even doable?
   Choices:
   0 = One person had a pretty beautiful hat.;
   1 = The host had a hat himself and he forget to count it.;
   2 = One person had moved away from the group when the host was counting.;
   3 = None of above.;
   Response: If one person had a pretty beautiful hat, then the host would still count 30 hats. If one person had moved away from the group when the host was counting, then the host would still count 30 hats. The host had a hat himself and he forget to count it. The answer is 1

7. Question: A certain kind of animal has parents but no children, lives happily but can not give birth to offspring. This kind of animal has existed on earth for a long history. How is that possible?
   Choices:
   0 = The animal is the dinosaur. Millions of years ago, dinosaurs roamed the Earth, but eventually, they became extinct due to various factors such as climate change or asteroid impact.;
   1 = The animal is the butterfly. During the

pupal stage of their life cycle, butterflies are inside a protective casing, and it may appear as if they have completely disappeared or been wiped out.;
2 = The animal is the Mule. Since all Mules are born sterile, you can only get a Mule by crossing a donkey with a horse.;
3 = None of above.;
Response: Dinosaurs are no longer alive, so they cannot live happily. Butterflies can give birth to offspring, so they are not the answer. Mules are born sterile, so they cannot give birth to offspring. The answer is 2

8. Question: Six women were hiking on a trail when a sudden snowfall began. Five of the women hurried along, but the sixth did not. Still, they all reached the cabin at the same time, and all but the sixth were cold. How is that possible?
Choices:
0 = The 6th woman beated them at the end.;
1 = The sixth woman was in a photograph that one of the others was carrying.;
2 = The 6th woman walked slowly to dodge the snow.;
3 = None of above.;
Response: If the sixth woman beat them or walked slowly to dodge the snow she would not have reached the cabin at the same time. If the sixth woman was in a photograph that one of the others was carrying, then she would not be cold and would not have to hurry. The answer is 1

B New CoT Prompts

B.1 New Base Prompts

1. Question: A couple is having a disagreement over the man's error. The man kept apologizing and pleading with the woman to allow him to see her directly. The woman was still upset. Thus, she wouldn't agree. The couple, though, was positioned on the same mat. How is that even doable?
Choices:
0 = The mat was folded in multiple times and end up with a special shape.;
1 = The woman kept turning her back to the man.;
2 = The couple was standing on two sides of the girl's apartment door, which had a rug under it.;
3 = None of above.;
Response: The couple are standing on the same mat but cannot see each other, which suggests that there is a barrier between them. Regardless of how the mat was folded, it would not prevent the couple from seeing each other. Even if the woman turned her back to the man he would still see her. Therefore, the couple was standing on two sides of a door, which had a rug under it. The answer is 2

2. Question: Eight people were sitting under a large tree. Suddenly, a gust of wind blows, yet none of them got hit by any falling leaves. How is this possible?
Choices:
0 = It was winter and the tree doesn't have any leaves.;
1 = People were camping under the tree.;
2 = The wind blows heavily.;
3 = None of above.;
Response: If the people were camping under the tree, they would still be hit by the falling leaves. The wind blowing heavily would cause the leaves to fall. If it was winter and the tree doesn't have any leaves, then the people would not be hit by any falling leaves. The answer is 0

3. Question: The ship was in the central Pacific Ocean. The ship suddenly began to sink without being crushed. However, each team was still preoccupied with its own tasks, so none was threatening. Why?
Choices:
0 = An underwater earthquake caused a rapid drop in the water level and resulted in the loss of buoyancy for the ship.;
1 = It was a Submarine.;
2 = There are too many fish around the ship.;
3 = None of above.;
Response: An earthquake would cause people to feel threatened, and fish around the ship would not matter. If the ship were a submarine, it would be designed to sink. The answer is 1

4. Question: Two mothers and two daughters were asking for new state IDs, but the agent only gave out three forms and instructed them on how to fill them out. Why?
Choices:
0 = They are one daughters, one mother and

one grandmother.;
1 = Two girls filled the from together.;
2 = One mother is too old to apply for new IDs.;
3 = None of above.;
Response: The agent only gave out three forms and instructed them on how to fill them out, which suggests that there are only three people. If they are one daughter, one mother, and one grandmother, then there would be three people. The answer is 0

5. Question: Every night, a man would sleep with a light on, as bright as the sun, dazzling the neighbors. But why did his neighbors never complain?
Choices:
0 = All his neighbors are blind.;
1 = The man would wake up early.;
2 = He lived in a light house.;
3 = None of above.;
Response: If all his neighbors are blind, they would not be dazzled by the light. If the man lived in a light house then the light would be expected. The answer is 2

6. Question: Each of the 30 participants in the masquerade had to wear a unique hat to distinguish themselves from one another. The host, however, only tallied 29 when he counted the number of hats to determine attendance. All 30 persons had signed their names on the spreadsheet, which confused him. He repeated the count. There are still 29. How is that even doable?
Choices:
0 = One person had a pretty beautiful hat.;
1 = The host had a hat himself and he forget to count it.;
2 = One person had moved away from the group when the host was counting.;
3 = None of above.;
Response: If one person had a pretty beautiful hat, then the host would still count 30 hats. If one person had moved away from the group when the host was counting, then the host would still count 30 hats. The host had a hat himself and he forget to count it. The answer is 1

7. Question: A certain kind of animal has parents but no children, lives happily but can not give birth to offspring. This kind of animal has existed on earth for a long history. How is that possible?
Choices:
0 = The animal is the dinosaur. Millions of years ago, dinosaurs roamed the Earth, but eventually, they became extinct due to various factors such as climate change or asteroid impact.;
1 = The animal is the butterfly. During the pupal stage of their life cycle, butterflies are inside a protective casing, and it may appear as if they have completely disappeared or been wiped out.;
2 = The animal is the Mule. Since all Mules are born sterile, you can only get a Mule by crossing a donkey with a horse.;
3 = None of above.;
Response: Dinosaurs are no longer alive, so they cannot live happily. Butterflies can give birth to offspring, so they are not the answer. Mules are born sterile, so they cannot give birth to offspring. The answer is 2

8. Question: Six women were hiking on a trail when a sudden snowfall began. Five of the women hurried along, but the sixth did not. Still, they all reached the cabin at the same time, and all but the sixth were cold. How is that possible?
Choices:
0 = The 6th woman beated them at the end.;
1 = The sixth woman was in a photograph that one of the others was carrying.;
2 = The 6th woman walked slowly to dodge the snow.;
3 = None of above.;
Response: If the sixth woman beat them or walked slowly to dodge the snow she would not have reached the cabin at the same time. If the sixth woman was in a photograph that one of the others was carrying, then she would not be cold and would not have to hurry. The answer is 1

B.2 New Mixed Prompts

1. Question: A horse is tied to a five-meter rope in front of an old saloon. Ten meters behind the horse is a bale of hay. Without breaking his rope, the horse is able to eat the hay whenever he chooses. How is this possible?
Choices:
0 = The rope stretches proportionally, provid-

ing the extra length needed for the horse to reach the hay ten meters away.;
1 = The rope is not tied to anything else.;
2 = The walls of the saloon retract or collapse inwards, creating more space for the horse to reach the hay.;
3 = None of above.;
Response: That the rope is not tied to anything else is the simplest choice. The horse can reach the hay whenever he chooses. The answer is 1

2. Question: A woman who lives in new york legally married three men, she did not get divorce, get an enrollment, or legally seperate. How is this possible?
Choices:
0 = The woman is not a good person.;
1 = His husband is not a good husband.;
2 = She is a minister.;
3 = None of above.;
Response: In new york it is not legal to be married to more than one person at a time. Ministers perform marriages, they do not get married. The answer is 2

3. Question: Brad began by entering the office tower's 22nd story through the filthy common window. He slid the window open and leapt through it after becoming depressed. Outside the building, there was a cliff-like drop to the ground. Amazingly, he came to rest entirely unharmed. How could he have survived the fall if there was nothing to slow or cushion his descent?
Choices:
0 = Brad was so sick and tired of window washing, he opened the window and jumped inside.;
1 = The ground outside the building is wet.;
2 = Consistent exercise has made him a very strong man.;
3 = None of above.;
Response: A person cannot jump out of a 22 story building without injury. Brad must have jumped into the building. The answer is 0

4. Question: Danny had just passed under an overpass in his semi when he abruptly came to a stop. Danny accidentally drove under the overpass that was only just tall enough for his truck because he wasn't paying enough attention. He was unable to move forward or backward in the semi due to how tightly it was wedged. When another tracker passed by, he was told how simple it would be to remove the semi from underneath the bridge. What did he recommend?
Choices:
0 = He told Danny to left the bridge.;
1 = He told Danny to overturn the track.;
2 = He told Danny to let some air out of his tires.;
3 = None of above.;
Response: Danny has to reduce the height of his truck to pass the bridge. Letting the air out of his tires reduces the truck's height. The answer is 2

5. Question: I excavate little caves and keep my gold and silver there. I also create gold crowns and silver bridges. They are the tiniest things you can imagine. Everyone will eventually require my assistance, but many people are reluctant to accept it. Why?
Choices:
0 = I am a dentist.;
1 = I am a thief.;
2 = I am a miner.;
3 = None of above.;
Response: Tiny crowns and bridges are dental work. People don't like going to the dentist. The answer is 0

6. Question: Each of the 30 participants in the masquerade had to wear a unique hat to distinguish themselves from one another. The host, however, only tallied 29 when he counted the number of hats to determine attendance. All 30 persons had signed their names on the spreadsheet, which confused him. He repeated the count. There are still 29. How is that even doable?
Choices:
0 = One person had a pretty beautiful hat.;
1 = The host had a hat himself and he forget to count it.;
2 = One person had moved away from the group when the host was counting.;
3 = None of above.;
Response: The host is the one counting the hats. The host had a hat himself and he forget to count it. The answer is 1

7. Question: A certain kind of animal has par-

ents but no children, lives happily but can not give birth to offspring. This kind of animal has existed on earth for a long history. How is that possible?
Choices:
0 = The animal is the dinosaur. Millions of years ago, dinosaurs roamed the Earth, but eventually, they became extinct due to various factors such as climate change or asteroid impact.;
1 = The animal is the butterfly. During the pupal stage of their life cycle, butterflies are inside a protective casing, and it may appear as if they have completely disappeared or been wiped out.;
2 = The animal is the Mule. Since all Mules are born sterile, you can only get a Mule by crossing a donkey with a horse.;
3 = None of above.;
Response: The animal is still alive, has parents, but cannot have children. Mules are born sterile, so they cannot give birth to offspring. The answer is 2

8. Question: Six women were hiking on a trail when a sudden snowfall began. Five of the women hurried along, but the sixth did not. Still, they all reached the cabin at the same time, and all but the sixth were cold. How is that possible?
Choices:
0 = The 6th woman beated them at the end.;
1 = The sixth woman was in a photograph that one of the others was carrying.;
2 = The 6th woman walked slowly to dodge the snow.;
3 = None of above.;
Response: All the women hurried except the sixth and were cold. The sixth woman is not physically present. She must be in a photograph one of the others was carrying. The answer is 1

C System and User Prompts

"role": "system", "content": "You are a Question Answering Model that answers questions by finding logical entailments between the question and answer choices."

D Problematic CR Questions

1. SP-120_CR: Mark was in a playground where somebody noticed a great player playing and with the announcements, gathered a lot of people. There were many great players from basketball, volleyball, football, and even swimmers, But Mark directly went to the footballer and took a photo with him. How did he know who was the person that people got excited for in the first place?
"Since the playground was a football playground and the other players could've not been playing in the playground at the time of the announcement.",
"Since Mark was a crazy fan of football, only a football player can be considered as great player in his mind.",
"Since Mark stood closest to the football players, he only focused on football players and didn't notice others.",
"None of above."

2. SP-30_CR: Why do old people consume more food than young people.
"Older adults may have specific dietary requirements to address age-related issues",
"Older people require increased nutrient intake to support overall health and well-being.",
"Because older people live longer.",
"None of above."

3. SP-184_CR: Five people were at a football match, and a sudden shower started. The four that rushed to take cover still got soaked, but the one who didn't move stayed completely dry. Why didn't he get wet?
"The man is an excellent football player that can avoid rain in high speed."
"The man was lucky enough to avoid all the rain.
"He was a photograph, the other people were there to honor a former player.",
"None of above."

4. SP-166_CR: A farmer has 11 sheep. Half of them are white. How is this possible?
"One sheep is regarded as both white and other colors same time.",
"A famer raises his sheep in both white way and another way.",
"They are all white.",
"None of above."

5. SP-156_CR: A driver is told to make a stop every half an hour for the engine to cool down, for five times. How long do the stops take?
   "Three hours."
   "Two and a half hours.",
   "Two hours.",
   "None of above."

Zero Shot is All You Need at SemEval-2024 Task 9: A study of State of the Art LLMs on Lateral Thinking Puzzles

Erfan Moosavi Monazzah* and **Mahdi Feghhi***
Iran University of Science and Technology
moosavi_m, feghhi_me@comp.iust.ac.ir

Abstract

The successful deployment of large language models in numerous NLP tasks has spurred the demand for tackling more complex tasks, which were previously unattainable. SemEval-2024 Task 9 introduces the brainteaser dataset that necessitates intricate, human-like reasoning to solve puzzles that challenge common sense. At first glance, the riddles in the dataset may appear trivial for humans to solve. However, these riddles demand lateral thinking, which deviates from vertical thinking that is the dominant form when it comes to current reasoning tasks. In this paper, we examine the ability of current state-of-the-art LLMs to solve this task. Our study is diversified by selecting both open and closed source LLMs with varying numbers of parameters. Additionally, we extend the task dataset with synthetic explanations derived from the LLMs' reasoning processes during task resolution. These could serve as a valuable resource for further expanding the task dataset and developing more robust methods for tasks that require complex reasoning. All the codes and datasets are available in paper's GitHub repository[1].

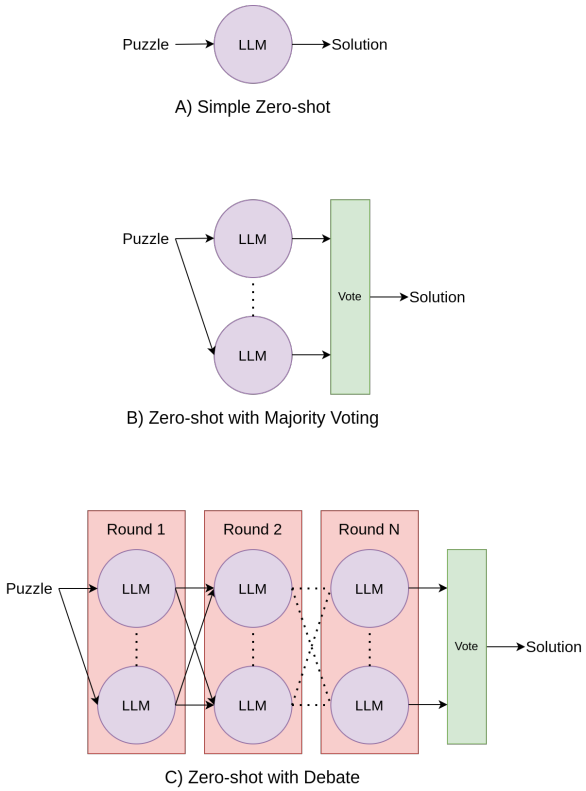


Figure 1: Different zero-shot configurations are shown. Figure A depicts the simple zero-shot usage of a Large Language Model. Figure B depicts the application of majority voting to a pool of LLMs. Figure C depicts a debate among a pool of LLMs over multiple rounds.

1 Introduction

In the domain of cognitive science, human reasoning is characterized by two distinct processes housed within the brain: 1) Vertical thinking and 2) Lateral thinking (Waks, 1997). Vertical thinking, also known as linear, convergent, or logical thinking, is an analytical process that progresses in a sequential manner. It is rooted in rationality, logic, and established rules, and is typically associated with the left hemisphere of the brain. Conversely, lateral thinking, colloquially referred to as "thinking outside the box", is a divergent and creative process. It entails approaching a problem from a novel perspective and challenging pre-existing assumptions, and is linked with the right hemisphere of the brain (Jiang et al., 2023). To solve a vertical puzzle, the model could follow a linear solution path and provide a step-by-step reasoning for the solution. However, the model was unable to provide a step-by-step solution for solving the lateral puzzle. Instead, it offered a fresh perspective on the puzzle and explained why the answer might be unreasonable when considering common sense. With the expansion of the LLMs market and re-

[1] github.com/ErfanMoosaviMonazzah/SemEval2024-Task9-BRAINTEASER

*Authors contributed equally to this work

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1889–1893
June 20-21, 2024

	Model	SP	WP
Ours	**Bing***	**86.7**	**97.9**
	Gemini (Team et al., 2023)	70.8	77.1
	Mixtral 8x7B (Jiang et al., 2024a)	63.3	71.9
	ChatGPT (Brown et al., 2020)	62.5	71.9
	ChatGPT (Vote)	67.5	76.0
	ChatGPT (Debate)	65.0	83.3
	Phi-2 (Abdin et al., 2023)	29.2	47.9
Base Paper	ChatGPT	62.7	53.5
	FlanT5 XXL (Chung et al., 2022)	34.0	39.8
	T0PP (Sanh et al., 2022)	30.5	34.8
	Human	92.0	91.7
	Random	24.4	25.3

Table 1: Zero-Shot models benchmarks over Brainteaser dataset. Values are the accuracy of the model over the dataset. Our submission on leader board is the output of the Bing.

search community, the demand for more complex benchmarks and tasks has increased as previous tasks become trivial for each new generation of models. As proposed by (Jiang et al., 2023), examining the ability of current state-of-the-art LLMs to reason on lateral thinking puzzles is a crucial aspect of their reasoning capabilities, which is the primary focus of this paper. We investigate a set of LLMs in a single model zero-shot, multi-agent voting, and multi-agent debate setup (**?**). To diversify our experiments, we studied both state-of-the-art production-level LLMs like ChatGPT and Gemini, and open ones like Microsoft Phi-2 and Mixtral 8x7B. We also publish an extended brainteaser dataset, which contains the detailed answers from all the models and techniques we tested. This dataset could be used for further analysis of these models and help to expand this relatively small dataset into a comprehensive benchmark for lateral reasoning evaluation. Interestingly, we found that current state-of-the-art models, despite having errors when it comes to lateral thinking, show very promising results. Even smaller LLMs like Phi-2 were able to demonstrate lateral thinking procedures in some cases. We submitted our best results which are produced by Microsoft Bing Chat to the task leader-board and ranked 2nd and 10th for Word Puzzle and Sentence Puzzle, respectively. (Jiang et al., 2024b)

2 System Overview

2.1 Zero-Shot

(Jiang et al., 2023) frames the puzzles as multiple-choice question-answering tasks. Each question or riddle is accompanied by four potential answers, with the last option always being "None of the above". Among the remaining three choices, one is invariably correct, while the others are guaranteed to be incorrect. Empirical experiments reveal a significant challenge with the current formulation, specifically the last choice. For most of these riddles, the solution is not a logical expression but often involves wordplay or clever misconceptions. Due to these factors, LLMs tend to favor the last choice *"None of the above"* since none of the other answers make sense in the conventional vertical thinking manner. For example when we asked Mixtral model, *"What animal keeps the best time? 1) A watchdog. 2) An elephant. 3) A whale. 4) None of above."* the model responded *"The correct answer is 'None of the above'. The reasoning behind this is that animals do not keep time in the sense of telling the exact hour and minutes like a clock or watch does. However, some animals such as certain bird species and insects are known for their keen sense of timing when it comes to activities such as migration and hibernation."*. Moreover, this option comprises a negligible portion of the Train and Validation dataset. Assuming that this is also the case for the Test dataset (since the la-

Zero-shot Prompt	Which option is the answer to this riddle, explain in a step by step manner:
	What clothing does a house wear?
	1) Address.
	2) Suit.
	3) Wedding dress.
	please place your answer in a json format:
	{
	"explanation":
	"option_number":
	}

Model Response	{
	"explanation": "The phrase 'what clothing does a house wear?' is intentionally misleading because houses do not physically wear clothes. The answer lies within option (1): an address. In essence, you can think of an address as the 'clothing' a house wears since it identifies and distinguishes each building from another.",
	"option_number": 1
	}

Table 2: A sample prompt provided to the model and the corresponding response generated by the model.

bels were not available at the time), we modify the formulation and remove the last option to compel the models to provide an answer to the riddle. We prompt the models with this new formulation and request them to "solve the riddle", explicitly asking for an explanation of the reasoning procedure. To simplify the evaluation of their answers, we also ask them to provide their answers in JSON format which make it easier to parse their responses. A sample prompt and model response is illustrated in Table 2.

2.2 Majority Voting

Different answers can be extracted from the same model while sending the same prompt by merely adjusting the model generation temperature to high values. This technique allows us to create a pool of models, pose the exact same question to them, and expect to receive different answers. Having a set of answers for the same question enables us to conduct a vote among them and select the most voted answer. To prevent a tie between the three choices in each question, we pose the same question to four language models with different temperatures. This approach ensures that one choice has at least one more vote than the others. Although asking the same question from the same LLM with different temperatures multiple times increases the chance of getting the right answer, it also increases the cost and resource usage, especially in the case of production LLMs like ChatGPT.

2.3 Debate

In majority voting, each LLM is independent from the others that are trying to solve the same question. But what if we group them together and attempt to solve the problem in a debate? Specifically, we define a finite number of debate rounds. In the first round, we pose the same question to each model, and they begin to answer the question individually (exactly the same as in majority voting). Having the initial responses at hand, in each subsequent round, we can ask the model to reconsider its reasoning based on the response (both answer and reasoning) of other LLMs. This may help the current LLM to identify any flaws in its solution (if any flaws are present) and attempt to correct it (Du et al., 2023). The debate continues until all the LLMs at the debate table reach a consensus or a maximum number of rounds pass. In that case, majority voting will be used to determine the final answer of the debate.

3 Experimental Setup

We carry out experiments with five distinct LLMs. For more information about each of them, refer to Table 1. We interact with these models through APIs, with the exception of Microsoft Phi-2, for which we utilize a Google Colab session equipped with a T4 GPU. In the Zero-Shot and Majority Voting experiments, each riddle is presented once to each model, and the answer is recorded. However, for the Debate experiment, we maintain a history of the conversation with the model, which serves as

	Sentence Puzzle				
Model	Original	Semantic	Context	Ori Sem	Overall
ChatGPT	56.1	52.4	51.8	43.9	53.5
Human	91.7	91.7	91.7	91.7	91.7
Bing (ours)	95	82.5	82.5	80	86.7
	Word Puzzle				
Model	Original	Semantic	Context	Ori Sem	Overall
ChatGPT	60.8	59.3	67.9	50.7	62.7
Human	90.7	90.7	94.44	90.7	92
Bing (ours)	100	100	93.8	100	97.9

Table 3: Detailed accuracy computation for each part of the dataset.

What was the result of the tiny lobster's math test?			The phrase 'sea plus' sounds similar to 'A+', a common grading symbol indicating excellent performance. However, since the subject of the question is a 'tiny lobster', it seems more appropriate to give a humorous response rather than a literal interpretation. Therefore, the best answer is likely 2) Sea-plus.
Very-bad	Sea-plus	Very-Good	
What kind of tree is most safe?			At first glance, this appears to be a botany-related question asking about different kinds of trees. However, the term 'safe' suggests another dimension to the query. Oaks were commonly used for building ships and houses, providing shelter from environmental elements. Thus, out of the provided options, oak seems to fit best.
Pine	Hemlock	Oak	
What is the capital in Washington?			This question plays with words related to the state of Washington and its capital city, Olympia. If we look at the word 'Washington', the letters N, T, and W appear as individual characters. Among those choices, the first letter N stands for the correct answer—Olympia, the capital of Washington State.
The letter N	The letter T	The letter W	

Table 4: There are different types of answers. The first type is when the model responds correctly. The second type is when the model attempts vertical thinking instead of lateral thinking and fails. The third type is when the model becomes confused and generates a nonsensical explanation.

additional context containing the model's previous answers and reasoning from earlier rounds. We explicitly instruct each model to provide its answer in JSON format. Due to lack of resources we could only test ChatGPT with voting and debating.

4 Results

The majority of our models outperformed the paper baselines, as shown in Table 1. This superior performance was not confined to Production Level models like Bing or Gmini. Surprisingly, the open model Mixtral performed on par with ChatGPT. Among the smaller models, Phi-2 performed commendably well, considering it only has 2.7 billion parameters compared to FlanT5 or T0PP, which have 11 billion parameters. It outperformed those models on the Word Puzzle. Bing also surpassed human performance on Word puzzles, as shown in Table 2. We observed that voting can positively impact accuracy. However, when it comes to debating, the results are less robust. Although it performs reasonably well on the Word puzzle, its performance deteriorated on the Sentence puzzle. During the inspection of the results, we encountered three types of answers. The first type is where the model understands that it's dealing with lateral thinking puzzles. Not only does it solve the puzzle correctly, but it also mentions something like 'The puzzle is a play on words,' which indicates that the model grasped the concept of the puzzle. In the second type, the model attempts a vertical thinking procedure and tries to solve the puzzle in a literal sense. It tries to assign an answer and justify it using complex logic. In the third type, the models were unable to come up with any good explanation. It seems they got confused by the nature of the puzzle and started to generate nonsense. See table 4.

5 Conclusion

Although there is still a gap between the accuracy of LLMs and humans when it comes to solving challenging puzzles that require lateral thinking, they currently perform well considering the difficulty of this task. Our results indicate a promising path for using an ensemble of large language models to collaborate and solve a problem together, whether they are fine-tuned for this collaboration, like Mixtral, or we use prompting ideas like voting or debating. We believe this path still requires thorough research, specifically in the quality of the

reasonings generated by each model.

References

Marah Abdin, Jyoti Aneja, Sebastien Bubeck, and ... 2023. Phi-2: The surprising power of small language models.

Tom B. Brown, Benjamin Mann, Nick Ryder, and ... 2020. Language models are few-shot learners.

Hyung Won Chung, Le Hou, Shayne Longpre, and ... 2022. Scaling instruction-finetuned language models.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2024b. Semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1996–2010, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, and ... 2022. Multitask prompted training enables zero-shot task generalization.

Gemini Team, Rohan Anil, Sebastian Borgeaud, and ... 2023. Gemini: A family of highly capable multimodal models.

Shlomo Waks. 1997. Lateral thinking and technology education. *Journal of Science Education and Technology*, 6(4):245–255.

Edinburgh Clinical NLP at SemEval-2024 Task 2: Fine-tune your model unless you have access to GPT-4

Aryo Pradipta Gema[1]* **Giwon Hong**[1]* **Pasquale Minervini**[1] **Luke Daines**[2]
Beatrice Alex[3,4]
[1]School of Informatics, University of Edinburgh [2]Usher Institute, University of Edinburgh
[3]Edinburgh Futures Institute, University of Edinburgh
[4]School of Literatures, Languages and Cultures, University of Edinburgh
{aryo.gema, giwon.hong, luke.daines, p.minervini, b.alex}@ed.ac.uk

Abstract

The NLI4CT task assesses Natural Language Inference systems in predicting whether hypotheses entail or contradict evidence from Clinical Trial Reports. In this study, we evaluate various Large Language Models (LLMs) with multiple strategies, including Chain-of-Thought, In-Context Learning, and Parameter-Efficient Fine-Tuning (PEFT). We propose a PEFT method to improve the consistency of LLMs by merging adapters that were fine-tuned separately using triplet and language modelling objectives. We found that merging the two PEFT adapters improves the F1 score (+0.0346) and consistency (+0.152) of the LLMs. However, our novel methods did not produce more accurate results than GPT-4 in terms of faithfulness and consistency. Averaging the three metrics, GPT-4 ranks joint-first in the competition with 0.8328. Finally, our contamination analysis with GPT-4 indicates that there was no test data leakage.[1]

1 Introduction

Extracting insights from Clinical Trial Reports (CTRs) is vital for advancing personalised medicine, yet manual analysis of these vast datasets is impractical. The Natural Language Inference for Clinical Trial Data (NLI4CT) task (Jullien et al., 2024)[2] addresses this challenge by evaluating Natural Language Inference (NLI) systems' ability to understand and reason within this domain.

In this study, we evaluate various LLMs, such as LLaMA2 (Touvron et al., 2023b), Mistral (Jiang et al., 2023), MistralLite (Yin Song and Chen Wu and Eden Duthie, 2023), and GPT-4 (OpenAI, 2023). We employed prompting strategies like In-context Learning (ICL) and Chain-of-Thought (CoT) to improve their accuracy. We also proposed a Parameter-Efficient Fine-Tuning (PEFT) method that merges independently fine-tuned adapters trained with distinct objectives, namely a triplet loss and a language modelling (LM) loss, to improve the consistency of the LLMs.

Our findings reveal that our novel PEFT method improves the F1 and consistency scores of the LLMs. However, GPT-4 produces more accurate results than all of the models we considered, co-leading the competition leaderboard. Although GPT-4 places fifth in the F1 score, its high faithfulness and consistency scores highlight its potential for a reliable prediction in the clinical domain. Lastly, we conduct a contamination analysis of GPT-4 to check whether instances of the NLI4CT dataset were included in GPT-4's pre-training data.

2 Background

2.1 Task overview

The NLI4CT task leverages a collection of CTRs and expert-annotated hypotheses. This iteration places a heightened emphasis on faithfulness (robustness to semantic changes) and consistency (stability against semantic preserving alterations). Aside from this focus, the composition of the dataset and the task objective remains identical to the previous iteration (Jullien et al., 2023a,b). Table 1 contains statistics for each data split, organised by sample, section, and label types.

Section Types Each CTR consists of four sections: "Eligibility criteria", "Intervention", "Results", and "Adverse events". Hypotheses are sentences claiming information in a CTR section.

Sample Types The task presents two sample types: "Single" and "Comparison". "Single" samples provide all relevant evidence within one CTR, while "Comparison" samples require cross-referencing information from two CTRs.

Task Objective The task objective is to classify the relationship between hypotheses and corre-

*These authors contributed equally to this work.

[1]Our code is available at https://github.com/EdinburghClinicalNLP/semeval_nli4ct.

[2]https://sites.google.com/view/nli4ct/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1894–1904

Split	Total	Sample Type		Section Type				Label Type	
		Single	Comparison	Intervention	Eligibility	Results	Adverse Events	Ent.	Con.
Train	1,700	1,035	665	396	486	322	496	850	850
Dev	200	140	60	36	56	56	52	100	100
Test	5,500	2,553	2,947	1,542	1,419	1,235	1,304	1,841	3,659

Table 1: Dataset statistics of each split, categorised by sample, section, and label types.

sponding CTR(s) as "entailment" or "contradiction". "Entailment" implies that the hypothesis is supported by the CTR(s), while a "contradiction" classification suggests inconsistency.

2.2 Related work

LLMs demonstrated promising results in the medical domain. For example, Liévin et al. (2022) conducted evaluations on LLMs, including Codex (Chen et al., 2021) and InstructGPT (Ouyang et al., 2022) using zero-shot, few-shot, and CoT prompting. These LLMs show comprehension of complex medical questions, recall of domain knowledge, and nontrivial reasoning.

Despite the increasing use of general LLMs, domain adaptive fine-tuning remains a prevailing approach in the medical domain (Lehman et al., 2023). As LLMs continue to grow in size, PEFT gains preference over full-parameter fine-tuning due to its resource efficiency. Gema et al. (2023) proposed a two-stage PEFT framework, one for domain-adaptive pre-training and one for downstream fine-tuning, to adapt LLaMA (Touvron et al., 2023a) to the clinical outcome prediction tasks. Even though Gema et al. (2023) introduced the idea of combining multiple adapters, they did not explicitly merge the adapter weights. Chronopoulou et al. (2023) proposed AdapterSoup, which performs averaging of the weights of PEFT adapters trained on the same objective function and different domains to improve the model's performance.

Extending the adapter merging idea, we introduced a novel method to merge PEFT adapters that are trained on different training objectives: triplet loss and LM loss. We compared this method with strategies without parameter fine-tuning, such as zero-shot inference, ICL, and CoT.

3 System Overview

We experimented with two strategies. The first involved no fine-tuning, aiming to comprehend LLMs' inherent ability to solve clinical tasks. The second employed our proposed PEFT method to improve the consistency of the model. Both systems ingest CTR-hypothesis pairs, predicting the correct label one token at a time from left to right.

3.1 Without Parameter Fine-tuning

The system with no fine-tuning utilises the pre-trained general LLMs for prediction. We experimented with multiple prompting strategies:

Zero-shot Employing the LLMs without any fine-tuning and examples.

In-Context Learning (ICL) Adapting the LLMs by providing examples of how to perform a task. Due to the maximum context length of the LLMs, we limit experiments to two examples (2-shot).

Chain-of-Thought (CoT) Prompting LLMs with a phrase (e.g., *"Let's think step by step"*) (Kojima et al., 2022), encouraging a sequential reasoning.

ICL + CoT Adapting the LLMs with ICL examples that are augmented with reasoning steps.

Figure 1 shows the workflow of the system. Firstly, we prepare the ICL examples. The normal ICL strategy requires the CTR section, the hypothesis, and the true label. Meanwhile, the ICL+CoT strategy requires ICL examples with reasons. We use ChatGPT (`gpt-3.5-turbo-0613`) to generate reasoned ICL examples as it has demonstrated sufficient clinical understanding (Falis et al., 2024). Similar to He et al. (2023), We prompt ChatGPT with a phrase *"Reason the answer step by step"* along with the CTR section, statement, and true label from the training dataset. The true labels and generated explanations using the ICL strategy are then stored. See Appendix B.1 for ChatGPT's hyperparameters used for generating explanations.

Second, we retrieve the ICL examples using either a random or BM25 retriever. Random retriever fetches ICL examples randomly, while the BM25 retriever fetches the most similar training data to the hypothesis sentence in question. We skip this step if we do not intend to use ICL.

Third, we choose the prompt template. If CoT is not used, the ordinary prompt is employed. This prompt instructs LLMs to answer using only one word, either "Contradiction" or "Entailment". If

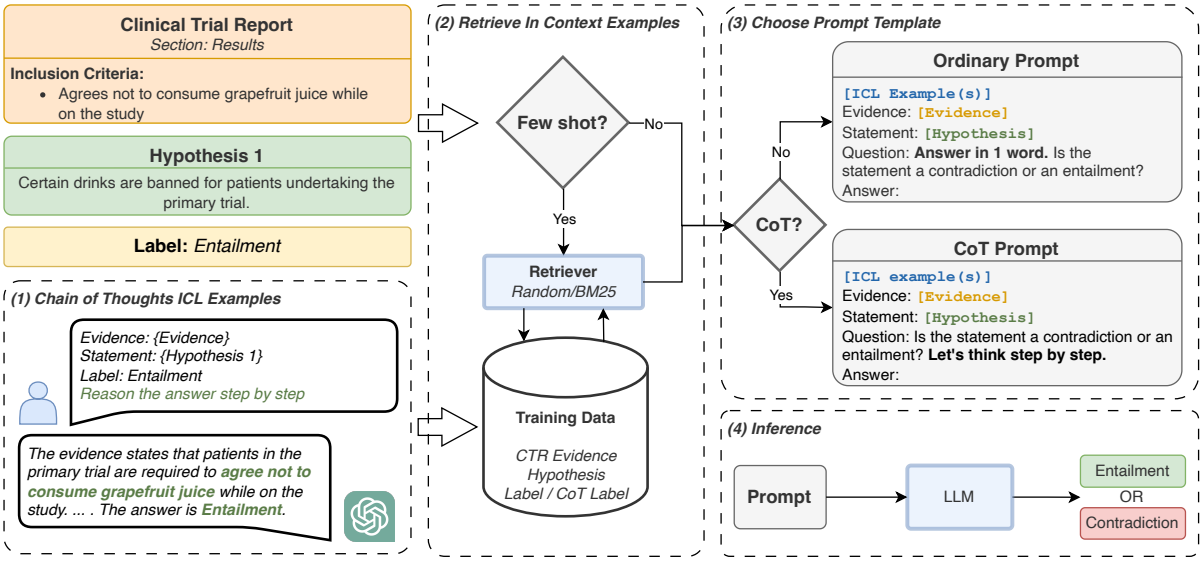


Figure 1: Our inference schema with multiple prompting strategies (without fine-tuning). For Chain-of-Thought examples, Natural Language Explanation was generated using ChatGPT (He et al., 2023).

CoT is used, the CoT prompt is used to instruct LLMs to think step by step. Refer to Figure 1.(3) and Appendix D for both final prompt designs.

Finally, the LLMs ingest the prompted input to generate an answer. To obtain the prediction, we checked which label appears last in the generated answer (either *"Entailment"* or *"Contradiction"*).

3.2 With Parameter Fine-tuning

We used LoRA (Hu et al., 2022) to fine-tune the parameters Φ_0 of a pretrained LLM $P_{\Phi_0}(y \mid x)$ on a training dataset $\mathcal{Z} = \{(x_i, y_i)\}_{i=1,\ldots,N}$. LoRA only trains a small number of additional parameters θ where $|\theta| \ll |\Phi_0|$; the parameters θ introduced by LoRA are used to define a new set of parameters Φ for the LLM, such that $\Phi = \Phi_0 + \Delta\Phi(\theta)$. The training objective for the additional parameters θ introduced by LoRA can be defined as:

$$\underset{\theta}{\operatorname{argmax}} \sum_{(x,y)\in\mathcal{Z}} f\left(P_{\Phi_0+\Delta\Phi(\theta)}\left(y \mid x\right)\right).$$

In our proposed method, we fine-tune two adapters using different training objectives, namely a Language Modelling objective (used to train the adapter parameters θ_{LM}) and a supervised learning objective based on the triplet loss (Balntas et al., 2016) (used to train the adapter θ_{triplet}).

In the supervised learning setting, we train LLMs using a triplet loss, with CTR serving as an anchor. Each CTR is associated with a pair of hypotheses, one contradiction and one entailment. The triplet loss encourages LLMs to map the entailment hypothesis closer to the CTR and the contradiction hypothesis to be far from the CTR.

$$L(a, p, n) = \max(0, d(a, p) - d(a, n) + \alpha),$$

where a, p, and n denote the averaged last hidden states of the LLM for the anchor (CTR), positive sample (entailment hypothesis), and negative sample (contradiction hypothesis), respectively. α is a margin.

We hypothesise that LM fine-tuning can improve the accuracy of the model on knowledge-intensive domain-specific downstream tasks, while supervised fine-tuning aids the model in distinguishing syntactically similar but semantically different data points and vice versa. Merging both adapters aims to achieve the best of both fine-tuning methods:

$$\theta_{\text{merged}} = \frac{1}{2}\left(\theta_{\text{LM}} + \theta_{\text{triplet}}\right).$$

This process resulted in one merged LoRA adapter, which can be re-attached to the original LLM. The base LLM, equipped with the merged LoRA, processes similarly prompted input, generating either *"Entailment"* or *"Contradiction"*. Refer to Figure 2 for an illustration of the workflow.

4 Results

The results shown in Table 2 can help us answer multiple research questions:

RQ 1: Can zero-shot LLMs perform well?

In a zero-shot setting, MistralLite-7B showed zero performance across all metrics due to it outputting

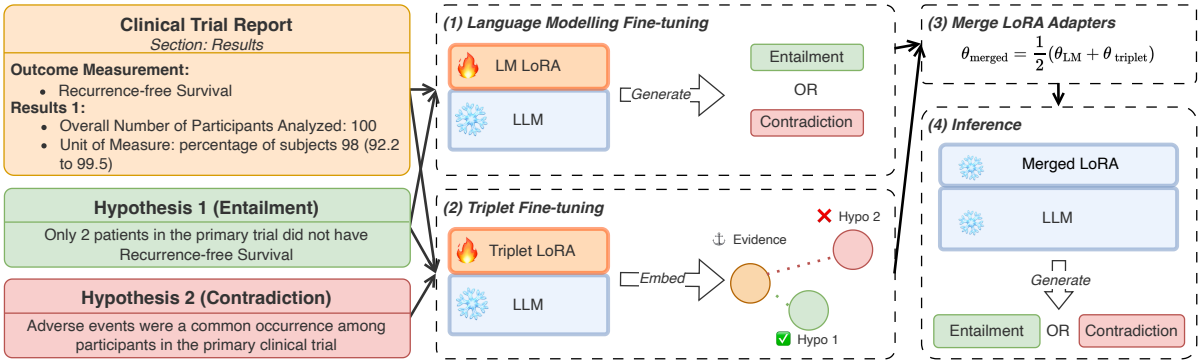


Figure 2: Our proposed fine-tuning scheme on SemEval 2024-Task 2. We suggested merging Adapters trained through Language Modelling (LM) Fine-tuning based on language modelling loss (in predicting either "Entailment" or "Contradiction") with Adapters trained through Triplet Fine-tuning based on triplet loss.

Model	F1	Faith.	Con.	Avg.
Mistral-7B-Instruct	0.6525	0.1343	0.4154	0.4007
+ 1-shot	0.6639	0.1111	0.4127	0.3959
+ 2-shot	0.6685	0.1343	0.4246	0.4091
+ CoT	0.4708	0.5926	0.5077	0.5237
+ CoT + 1-shot	0.5835	0.5706	0.5493	0.5678
+ CoT + 2-shot	0.5944	0.6065	0.5650	0.5886
MistralLite-7B	-	-	-	-
+ 1-shot	0.5389	0.4109	0.4826	0.4775
+ 2-shot	0.4665	0.6597	0.5413	0.5558
+ CoT	-	-	-	-
+ CoT + 1-shot	0.5628	0.4664	0.4973	0.5088
+ CoT + 2-shot	0.5801	0.4977	0.5164	0.5314
LLaMA2-7B-Chat	0.6417	0.1192	0.4159	0.3923
+ 1-shot	0.6451	0.1678	0.4376	0.4168
+ 2-shot	0.6308	0.1701	0.4304	0.4104
+ CoT	0.6369	0.3009	0.4775	0.4718
+ CoT + 1-shot	0.6101	0.3924	0.4855	0.4960
+ CoT + 2-shot	0.5607	0.4630	0.4925	0.5054
LLaMA2-13B-Chat	0.6069	0.4502	0.4940	0.5170
+ 1-shot	0.6303	0.3345	0.4882	0.4843
+ 2-shot	0.6169	0.4016	0.5012	0.5066
+ CoT	0.6028	0.5012	0.5116	0.5385
+ CoT + 1-shot	0.6346	0.5312	0.5360	0.5673
+ CoT + 2-shot	0.5919	0.6123	0.5549	0.5864
GPT-4	**0.7751**	**0.9479**	**0.7754**	**0.8328**

Table 2: Results on the test set across various LLMs with multiple prompting strategies (no fine-tuning).

an empty string. This suggests that, without any prompting strategies, it did not understand the given instruction. Mistral-7B-Instruct, LLaMA2-7B-Chat, and LLaMA2-13B-Chat show some degree of performance in the F1, faithfulness, and consistency metrics. Among the three, LLaMA2-13B-Chat achieved the highest faithfulness and consistency scores. GPT-4 stood out with the highest scores in all metrics, suggesting its strong performance even without any prompting strategies applied. This begs the question of whether any prompting strategies can be applied to help the relatively smaller LLMs perform better.

RQ 2: Can smaller LLMs perform on par with GPT-4 with prompting strategies?

In-Context Learning We investigated 1- and 2-shot settings using BM25. 1-shot setting consistently improved the performance of the LLMs (see Appendix C comparing random and BM25 ICL examples). With an ICL example, MistralLite-7B understood how to answer the prompted input. Mistral-7B-Instruct, LLaMA2-7B-Chat, and LLaMA2-13B-Chat also showed performance improvement compared to the zero-shot setting, albeit marginal. The 2-shot setting did not improve the LLMs consistently. Mistral-7B-Instruct showed an improvement in all metrics with 2-shot settings, while the other LLMs see F1 score drops, albeit the faithfulness and consistency may be improved.

Chain-of-Thought We investigated CoT in a zero-shot setting. Similarly to the zero-shot setting, MistralLite-7B showed zero performance in all metrics due to outputting an empty string. We saw drops in F1 scores for Mistral-7B-Instruct, LLaMA2-7B-Chat, and LLaMA2-13B-Chat, and improved the faithfulness and consistency scores. This indicates the efficacy of CoT in ensuring faithful and consistent answers from LLMs, albeit it may marginally harm the accuracy of the model.

In-Context Learning + Chain-of-Thought Since ICL improves the LLMs' F1 score, and CoT improves the faithfulness and consistency scores, we investigated the combination of both. The results show that ICL + CoT improves LLMs across metrics. Considering the averaged score,

2-shot ICL and CoT improve all LLMs except for MistralLite-7B.

Despite employing these strategies, the LLMs could not outperform GPT-4, particularly in terms of faithfulness and consistency. This suggests that while combining ICL and CoT is beneficial, it is still challenging to achieve parity with GPT-4.

RQ 3: Can fine-tuned smaller LLMs perform on par with GPT-4?

Model	F1	Faith.	Con.	Avg.
Mistral-7B-Instruct	**0.7689**	0.7662	0.7140	0.7497
MistralLite-7B	0.7478	0.8727	**0.7220**	**0.7808**
LLaMA2-7B-Chat	0.6073	0.7176	0.6146	0.6465
LLaMA2-13B-Chat	0.6766	0.7731	0.6610	0.7036
Meditron-7B	0.1980	**0.9560**	0.6165	0.5902

Table 3: Results on the test set across various LLMs with parametric-efficient fine-tuning.

As we may have reached the limit of performance using prompting strategies, we investigated employing fine-tuning the smaller LLMs.

Can LoRA fine-tuning improve the performance of LLMs? Table 3 presents the performance for each LLM fine-tuned with LoRA. Notably, fine-tuning leads to improvements across all metrics for all LLMs. MistralLite-7B is the best-performing LLM after fine-tuning with 0.7808 averaged scores, and it is notably better in terms of faithfulness and consistency scores compared to the other models. The fine-tuned Meditron-7B did not show a satisfactory overall performance. The subsequent experiment in merging LoRA adapters will focus on using MistralLite-7B as the base model.

Model	F1	Faith.	Con.	Avg
MistralLite-7B				
+ θ_{LM}	0.7478	0.8727	0.7220	0.7808
+ Avg (θ_{LM}, θ_{triplet})	**0.7824**	**0.8391**	**0.7372**	**0.7862**

Table 4: Results on the test set with our proposed merging adapters fine-tuning.

Can merging LoRA adapters improve the performance of LLMs? Table 4 displays results obtained through fine-tuning MistralLite-7B with only LM adapter θ_{LM} and the average of θ_{LM} and θ_{triplet} adapters. The merged θ_{LM} and θ_{triplet} adapters improve the overall performance of the LLM (joint-fourth in the competition). It achieves a better F1 score of 0.7824 (+0.0346), indicating that merging LoRA adapters may improve the predictive performance of LLMs. We noticed a lower faithfulness score (-0.0336) and a higher consistency score (+0.0152). This indicates the model struggles to understand semantic changes introduced by deliberate alterations but can understand semantically similar data better.

4.1 Contamination Analysis on GPT-4

Inspired by Carlini et al. (2022), we assessed whether instances of the NLI4CT dataset were included in GPT-4's pre-training data. We prompted GPT-4 with: 1) System instruction: *"You are a helpful assistant on the SemEval task. Complete the given statement."*, 2) Truncation of half of the statement to prompt GPT-4 to infer the remaining. (refer to Appendices B.7 and D.3 for details)

We define two metrics: *extractable match*, checking if the predicted half of the statement by GPT-4 is included in the original half, and *partial match*, assessing how sequentially each token of the predicted half of the statement is included in the original half. In the test set, GPT-4 recorded an extractable match score of 0.033 and a partial match score of 0.322. The low extractable match score may indicate that GPT-4 has not seen the test data during its pretraining, whereas the higher partial match score may indicate GPT-4's ability to identify keywords from CTRs.

5 Conclusion

This study assesses the performance of various LLMs, employing diverse strategies such as CoT, ICL, and PEFT. We propose a PEFT method, merging independent adapters fine-tuned separately using triplet and LM losses. Our proposed PEFT method improves the F1 and consistency scores but reduces faithfulness — our best fine-tuned model, MistralLite-7B + LM LoRA + Triplet LoRA, achieved an average score of 0.7862. However, it does not outperform GPT-4 in terms of faithfulness and consistency: GPT-4 ranks joint-first in the competition with an average score of 0.8328. A contamination analysis on GPT-4 revealed no NLI4CT test data leakage, indicated by a low extractable match score (0.033), and showcased its ability to identify keywords from CTRs with a relatively high partial match score (0.322).

Limitations

Due to the scope of the study and the limited resources, we opted to only experiment with GPT-4 in a zero-shot setup. However, our proposed strategies that improved the performance of smaller LLMs could also be used to enhance GPT-4. Albeit the promising performance of the LLMs, particularly GPT-4, the predictions may still be inaccurate and should not be used in a clinical setting without human supervision.

We conducted a contamination analysis inspired by Carlini et al. (2022) and concluded that there may be no test data leakage during the pretraining of GPT-4. However, we acknowledge that contamination analysis alone may not be sufficient in proving test data leakage.

Acknowledgements

APG was supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics. PM was partially funded by ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence), EPSRC (grant no. EP/W002876/1), an industry grant from Cisco, and a donation from Accenture LLP; and is grateful to NVIDIA for the GPU donations. BA was partially funded by Legal and General PLC as part of the Advanced Care Research Centre and by the Artificial Intelligence and Multimorbidity: Clustering in Individuals, Space and Clinical Context (AIM-CISC) grant NIHR202639. For the purpose of open access, The authors have applied a Creative Commons attribution (CC BY) licence to any author-accepted manuscript version arising. Experiments from this work are conducted mainly on the Edinburgh International Data Facility[3] and supported by the Data-Driven Innovation Programme at the University of Edinburgh.

References

Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*. BMVA Press.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Alexandra Chronopoulou, Matthew E Peters, Alexander Fraser, and Jesse Dodge. 2023. Adaptersoup: Weight averaging to improve generalization of pretrained language models. *arXiv preprint arXiv:2302.07027*.

Matúš Falis, Aryo Pradipta Gema, Hang Dong, Luke Daines, Siddharth Basetti, Michael Holder, Rose S Penfold, Alexandra Birch, and Beatrice Alex. 2024. Can gpt-3.5 generate and code discharge summaries? *arXiv preprint arXiv:2401.13512*.

Aryo Pradipta Gema, Luke Daines, Pasquale Minervini, and Beatrice Alex. 2023. Parameter-efficient fine-tuning of llama for the clinical domain. *arXiv preprint arXiv:2307.03042*.

Xuanli He, Yuxiang Wu, Oana-Maria Camburu, Pasquale Minervini, and Pontus Stenetorp. 2023. Using natural language explanations to improve robustness of in-context learning for natural language inference. *arXiv preprint arXiv:2311.07556*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

[3] https://edinburgh-international-data-facility.ed.ac.uk/

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Eric Lehman, Evan Hernandez, Diwakar Mahajan, Jonas Wulff, Micah J Smith, Zachary Ziegler, Daniel Nadler, Peter Szolovits, Alistair Johnson, and Emily Alsentzer. 2023. Do we still need clinical language models? *arXiv preprint arXiv:2302.08091*.

Valentin Liévin, Christoffer Egeberg Hother, and Ole Winther. 2022. Can large language models reason about medical questions? *arXiv preprint arXiv:2207.08143*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yin Song and Chen Wu and Eden Duthie. 2023. amazon/MistralLite.

Parameter	Value
Model Name	gpt-3.5-turbo-0613
API Version	2023-03-15-preview
Temperature	0
Top P	0
Frequency Penalty	0
Presence Penalty	0
Max new token	256
System Prompt	You are a helpful clinician's assistant designed to identify if a clinical statement is a contradiction or an entailment to the presented evidence.
Prompt	Evidence: [Evidence] Statement: [Statement] Question: Answer in 1 word. Is the statement a contradiction or an entailment? Answer: [Label] Reason the answer step by step

Table 5: Azure API call hyperparameters.

Hyperparameter	Value
Epoch	10
Gradient accumulation step	32
Optimiser	AdamW
Learning rate	0.001
Weight decay	0.01
Max sequence length	2048

Table 6: Language Modelling training hyperparameters.

Hyperparameter	Value
Epoch	10
Gradient accumulation step	32
Optimiser	AdamW
Learning rate	0.00001
Weight decay	0.01
Max sequence length	1024
Triplet loss margin	1.0
Triplet loss p	2
Triplet loss ϵ	1e-7

Table 7: Triplet training hyperparameters.

A Experimental setup

We use HuggingFace's Transformers (Wolf et al., 2020) and PEFT (Mangrulkar et al., 2022) libraries for the experiments. All inferences and fine-tuning experiments were run on two NVIDIA A100-40GB GPUs.

For models without parameter fine-tuning (prompting strategies, subsection 3.1), in-context examples were retrieved from the Training set (for both random and BM25 retrievers). Additionally, the Dev set was used to evaluate and select the optimal prompt design. Models with parameter fine-tuning (subsection 3.2) were trained using the Training set, and the Dev set was utilised to determine the best checkpoint.

B Hyperparameters

B.1 ChatGPT Hyperparameters for the generation of Natural Language Explanation

We prompted GPT-3.5 (model name: `gpt-3.5-turbo-0613`) with hyperparameters as shown in Table 5. The generation process took approximately 2 hours and cost $2.

B.2 GPT-4 generation hyperparameters

We prompted GPT-4 (model name: `gpt-4`) with the ordinary prompt as shown in Figure 1. We set `temperature=0` to ensure that the model's generation is deterministic. The maximum generation length is 8. The generation process took approximately 2 hours and cost $77.

B.3 Non GPT-4 generation hyperparameters

All models (apart from GPT-4) were loaded in BFloat16 to ensure that they fit into our resources. We used `do_sample=False` to ensure that the model's generation is deterministic. The maximum generation length is 8 new tokens for non-CoT experiments and 100 for CoT experiments.

B.4 Language Modelling training hyperparameters

LM training used the hyperparameters detailed in Table 6. The LLM's maximum sequence length is adjusted to fit on two NVIDIA A100-40GB GPUs.

B.5 Triplet training hyperparameters

Triplet training used the hyperparameters detailed in Table 7. The LLM's maximum sequence length is adjusted to fit on two NVIDIA A100-40GB GPUs. Triplet training demands more memory because we need to generate three hidden representations during training (i.e., anchor, positive, negative), necessitating a reduction in sequence length.

B.6 PEFT Hyperparameters

All LLMs and training methods (i.e., LM and triplet training) used the same LoRA hyperparameters as shown in Table 8.

Hyperparameter	Value
r	16
alpha	32
dropout	0.0
target_modules	["k_proj", "q_proj", "v_proj"]

Table 8: LoRA Hyperparameters.

Model	ICL	F1	Faith.	Con.	Avg.
Mistral-7b-Instruct	Random: 1-shot	0.6694	0.0856	0.4086	0.3879
Mistral-7b-Instruct	BM25: 1-shot	0.6639	0.1111	0.4127	0.3959
Mistral-7b-Instruct	Random: 2-shot	0.6639	0.1458	0.4294	0.4130
Mistral-7b-Instruct	BM25: 2-shot	0.6685	0.1343	0.4246	0.4091
MistralLite-7B	Random: 1-shot	0.6622	0.0150	0.3854	0.3542
MistralLite-7B	BM25: 1-shot	0.5389	0.4109	0.4826	0.4775
MistralLite-7B	Random: 2-shot	0.5097	0.5023	0.5164	0.5095
MistralLite-7B	BM25: 2-shot	0.4665	0.6597	0.5413	0.5558
LLaMA2-7B-Chat	Random: 1-shot	0.6613	0.0116	0.3864	0.3531
LLaMA2-7B-Chat	BM25: 1-shot	0.6451	0.1678	0.4376	0.4168
LLaMA2-7B-Chat	Random: 2-shot	0.6387	0.1250	0.4180	0.3939
LLaMA2-7B-Chat	BM25: 2-shot	0.6308	0.1701	0.4304	0.4104
LLaMA2-13B-Chat	Random: 1-shot	0.6585	0.3113	0.4724	0.4807
LLaMA2-13B-Chat	BM25: 1-shot	0.6303	0.3345	0.4882	0.4843
LLaMA2-13B-Chat	Random: 2-shot	0.6230	0.4074	0.4935	0.5080
LLaMA2-13B-Chat	BM25: 2-shot	0.6169	0.4016	0.5012	0.5066

Table 9: Comparison of In-Context Learning Models Using Random and BM25 Retrievers on the Test set

B.7 Contamination Analysis on GPT-4

For the Contamination Analysis, we utilised the same settings as those described in Appendix B.2, specifically setting the maximum number of generated tokens to 8. This was done to prevent the incorrect biases due to excessively lengthy predictions by GPT-4, as our evaluation method focuses on determining whether the prediction is included within the ground truth.

C Ablation study on Random vs Relevance-based In-Context Examples

We also compared the performance of the model by using random and relevant ICL examples. As shown in Table 9, we found that relevant ICL examples helped the LLMs achieve better faithfulness and consistency scores, while the F1 scores may be impacted. For that reason, we opted to use relevance-based ICL examples for the ICL-based runs.

D Prompt Examples

Here, we provide examples of the prompts used in our experiments. The black text within the box represents the prompt input text, the red text represents the prediction of the models, and the blue text represents the ground truth.

D.1 Base Prompt for Zero-shot and In-Context Learning

Base Prompt

You are a helpful assistant.

«In-Context Learning Examples»

Evidence:
Primary trial:
Adverse Events 1:
Total: 0/15 (0.00%)
Adverse Events 2:
Total:

Secondary trial:
Adverse Events 1:
Total: 0/442 (0.00%)
Adverse Events 2:

Statement: the primary trial and the secondary trial do not have any recorded adverse events for their participants. crypt is a pitlike depression or tubular recess.
Question: Answer in 1 word. Is the statement a contradiction or an entailment?
Answer: Entailment

Ground Truth: Entailment

D.2 Chain-of-Thought for Zero-shot and In-Context Learning

Chain-of-Thought

You are a helpful assistant.

«In-Context Learning Examples»

Evidence:
Primary trial:
Adverse Events 1:
Total: 0/15 (0.00%)
Adverse Events 2:
Total:

Secondary trial:
Adverse Events 1:
Total: 0/442 (0.00%)
Adverse Events 2:

Statement: the primary trial and the secondary trial do not have any recorded adverse events for their participants. crypt is a pitlike depression or tubular recess.
Question: Is the statement a contradiction or an entailment?
Let's think step by step
Answer: Great, let's analyze the statement and the evidence provided to determine if it's ... because the evidence shows that there are no adverse events recorded for the participants in either trial.
Therefore, the answer is Entailment."

Ground Truth: Entailment

D.3 Contamination Analysis on GPT-4

Extractable Match

You are a helpful assistant on the semeval task. Complete the given statement.

Evidence:
Primary trial:
Outcome Measurement:
Number of Participants With Reduction in CTCs Following High-dose Chemotherapy With Purged Autologous Stem Cell Products
Number of circulating tumor cells (CTCs) measured at one month post autologous hematopoietic stem cell transplantation (AHST), considered both as longitudinal values and compared to the baseline number of CTCs.
Time frame: Baseline to 1 month post AHST
Results 1:
Arm/Group Title: High-dose Chemotherapy
Arm/Group Description: Carboplatin + Cyclophosphamide + Thiotepa
Carboplatin : Target AUC of 20, then divided into 4 doses given by vein (IV) days -6, -5, -4, -3 prior to stem cell infusion.
Thiotepa : $120mg/m^2$ by vein days -6, -5, -4, -3 prior to stem cell infusion.
Stem Cell Transplant : Stem Cell Transplant on Day 0.
Cyclophosphamide : $1.5gm/m^2$ by vein days -6, -5, -4, -3 prior to stem cell infusion.
Overall Number of Participants Analyzed: 21
Measure Type: Number
Unit of Measure: participants 9
Statement: less than half of the primary trial participants had a Reduction in circulating tumor cells Following High-dose Chemotherapy With Pur

Ground Truth: Following High-dose Chemotherapy With Purged Autologous Stem Cell Products

Partial Match

You are a helpful assistant on the semeval task. Complete the given statement.

Evidence:
Primary trial:
Adverse Events 1:
Total: 3/12 (25.00%)
Hemoglobin 1/12 (8.33%)
Alkaline phosphatase 1/12 (8.33%)
Dehydration 1/12 (8.33%)
Syncope 2/12 (16.67%)
Dyspnea 1/12 (8.33%)
Hypotension 1/12 (8.33%)

Secondary trial:
Adverse Events 1:
Total: 0/115 (0.00%)
Deep vein thrombosis * [1]0/115 (0.00%)
Adverse Events 2:
Total: 1/119 (0.84%)
Deep vein thrombosis * [1]1/119 (0.84%)
Statement: on both the primary and secondary clinical trials, syncope was reported as an adverse event in the

Ground Truth: emerged as the most common adverse occurrence in the patient groups

CaresAI at SemEval-2024 Task 2: Improving Natural Language Inference in Clinical Trial Data using Model Ensemble and Data Explanation

Reem Abdel-Salam
Cairo University / Egypt
CaresAI/ Australia
reem.abdelsalam13@gmail.com

Mary Adetutu Adewunmi
University of Tasmania / Australia
CaresAI/ Australia
Mary.Adewunmi@utas.edu.au

Mercy Akinwale
Covenant University / Nigeria
CaresAI/ Australia
mercy.akowepgs@stu.cu.edu.ng

Abstract

Large language models (LLMs) have demonstrated state-of-the-art performance across multiple domains in various natural language tasks. Entailment tasks, however, are more difficult to achieve with a high-performance model. The task is to use safe natural language models to conclude biomedical clinical trial reports (CTRs). The Natural Language Inference for Clinical Trial Data (NLI4CT) task aims to define a given entailment and hypothesis based on CTRs. This paper aims to address the challenges of medical abbreviations and numerical data that can be logically inferred from one another due to acronyms, using different data pre-processing techniques to explain such data. This paper presents a model for NLI4CT SemEval 2024 task 2 that trains the data with DeBERTa, BioLink, BERT, GPT2, BioGPT, and Clinical BERT using the best training approaches, such as fine-tuning, prompt tuning, and contrastive learning. Furthermore, to validate these models, different experiments have been carried out. Our best system is built on an ensemble of different models with different training settings, which achieves an F1 score of 0.77, a faithfulness score of 0.76, and a consistency score of 0.75 and secures the sixth rank in the official leaderboard. In conclusion, this paper has addressed challenges in medical text analysis by exploring various NLP techniques, evaluating multiple advanced natural language models(NLM) models and achieving good results with the ensemble model. Additionally, this project has contributed to the advancement of safe and effective NLMs for analysing complex medical data in CTRs.

1 Introduction

Clinical trials play a crucial role in advancing medical knowledge, evaluating the safety and efficacy of new treatments, and improving patient care (Holford et al., 2010) which are essential for the development of new drugs, therapies, and medical interventions. Most importantly, they involve systematic investigations that aim to answer specific research questions and provide evidence-based guidance for medical decision-making (Tunis et al., 2003). Moreover, clinical trial reports (CTRs) have been published at an accelerated rate due to the rapid development of digital health. Currently, there are more than 10,000 CTRs just for breast cancer (Jullien et al., 2024; Bastian et al., 2010). Also, medical professionals have developed evidence-based clinical diagnoses through the increasing number of Clinical Trial Reports (CTRs) (Bastian et al., 2010), which serve as a broad source of factual and scientific information. Despite these CTRs, drawing valuable conclusions from these reports can be an uphill task due to the different medical domains and the unstructured nature of the report. Recent improvements in natural language processing (NLP) systems, on the other hand, have led to the idea of using multiple language models that have already been trained in the medical field to efficiently carry out medical NLP tasks. The growth of CTRs has also made it possible for a natural language inference (NLI) system to be created that can help with medical interpretation and finding evidence for individualized evidence-based therapy.(Agrawal et al., 2022) used InstructGPT with zero-shot and few-shot settings to extract information from clinical text. In addition, the authors introduced new datasets for benchmarking for few-shot clinical information extraction. The work in (Molinet et al., 2022) introduced a new tool, the ACTA automated tool, to support evidence-based clinical decision-making. The authors in (Yasunaga et al., 2022) proposed a new model, LinkBERT, that incorporates document link knowledge for medical domains. Despite substantial research on the use of advanced NLP approaches in the medical domain, evaluation benchmarks remain inadequate.

The SemEval 2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1905–1911

(NLI4CT) task is proposed by (Jullien et al., 2024) by building an efficient evaluation benchmark using a set of statements, explanations, and CTRs for breast cancer. This task is an extension of the previous year's shared task Multi-evidence Natural Language Inference for Clinical Trials. The purpose of the NLI4CT task is to entail a statement based on one or multiple clinical reports. NLI4CT is challenging because hypothesis verification sometimes requires integrating multiple pieces of data from the premise. In some instances, validating a hypothesis necessitates a comparison of two distinct premise CTRs. Validating hypotheses based on each premise type demands varying levels of inference skills (textual, numerical, etc.).

This paper presents work done in the NLI4CT to address these challenges owing to the complexity of the medical domain and text structure. The objective of this task is to develop a system capable of deducing conclusions or implications about various CTRs. The system consists of an ensemble of different experiments using different training approaches. The rest of the papers go as follows: section 3 discusses the proposed methods, section 4 shows experimental results, and section 5 concludes the paper.

2 Background

The main goal of the task is to determine the validity of a claim (hypothesis) based on a single section from one or multiple clinical trial reports (CTRs) of breast cancer (premises). There are two possible inferential relations for each statement: entailment and contradiction. The dataset[1] used is provided by the task organizers and it is divided into two parts: The first part is derived from a compilation of CTRs, which is categorized into four sections.: a) eligibility criteria required for participation in the clinical trial; b) intervention detailing the treatment type, dosage, frequency, and duration; c) results showing participant numbers, outcome measures, units, and findings; and d) adverse effects observed in patients during the trial. The second part compromises the claim about the information contained in a single section, either in one or two CTRs and information about which CTRs are targeted and which section. The dataset consists of 1700 training samples and 200 validation samples. The dataset supplied has an equal distribution of labels.

[1] https://github.com/ai-systems/Task-2-SemEval-2024

3 System overview

This section presents the various approaches used while developing the final models. This includes techniques for preprocessing and ingesting data. Moreover, it includes the paradigms used for training as well as the experimental setup. DeBERTa and Bert-based models are fine-tuned using a weighted ensemble of refined iterations, as well as prompt-based fine-tuning (Lester et al., 2021) for DeBERTa final models.

3.1 Data Preparation

Large Models (LM) have challenges doing reasoning tasks when presented with acronyms, numerical, and quantitative information. Large language models (LLM) can be used to answer numerical entities and improve performance through a chain of thought, which is a methodical process of reasoning and explaining things step by step (Wang et al., 2023a). We have implemented a comparable pre-processing procedure for converting numeric quantities into textual representations and providing explanations in the presence of mathematical formulas. Furthermore, abbreviations are expanded to their original full names. The process goes as follows: first numbers, operations and abbreviations are converted into textual representation. For example **"Arm/Group Description: Phase 1 Cyclophosphamide Dose Escalation: Fludarabine 5x25mg/m2 + Cyclophosphamide 2x60mg/kg + Cells intravenous (IV) "** is converted to **"Arm/Group Description: Phase one Cyclophosphamide Dose Escalation: Fludarabine five times twenty-five milligram/ meterstwo plus Cyclophosphamide two times sixty milligram/ kilograms plus Cells intravenous (IV)."** The next steps involve using LLM as chatGPT or Mistral 7b to give an explanation for the converted text. The following prompt is feed to the LLM ***" Explain the following sentences in five sentences: text".*** . ChatGPT provides the following explanation: " **The given sentence describes a specific group or category within a clinical trial referred to as "Arm/Group Description" during the phase one stage. This particular arm involves a treatment protocol called "Cyclophosphamide Dose Escalation," which aims to determine the optimal and safe dosage of Cyclophosphamide, a chemotherapy drug. The treatment**

regimen includes the intravenous administration (IV) of Fludarabine at a dose of twenty-five milligrams per square meter of body surface area, repeated five times over a specific time period or schedule. Additionally, Cyclophosphamide is administered intravenously (IV) at a dose of sixty milligrams per kilogram of body weight, repeated twice over the same time period or schedule. The treatment also involves the administration of cells intravenously (IV), although the specific type of cells is not mentioned in the provided sentence.". Finally, Both Converted text and explanation are concatenated.

3.2 Data Representation

In types of reasoning tasks such as hypothesis and premise reasoning, the intuitive step is to feed both texts separated by a model-separating token ***("hypothesis [SEP] premise")***. However, it is not yet clear if it is an optimal representation or not. To address this, different representations were used to assess model performance and sensitivity to data representations.

The following illustrates several data representations used. The challenge included two sorts of reasoning. The first type of reasoning is to check whether a given premise is correct based on one single report. The second type is to check whether a given premise is correct based on two different reports. Therefore the model was two different representations were designed for each type.

- The first common design is to feed in each premise section separated by a separator token, then the hypothesis section.
  - ***"First premise [SEP] Second premise [SEP] hypothesis".***
- The second design was adding special token information to indicate the following sections:
  - **token_first** for the first premise
  - **token_second** for the second premise
  - **token_hypothesis** for the hypothesis.
  - ***"token_first_section First premise [SEP] token_second_section Second premise [SEP] token_hypothesis".***
- The third design was inverting order first feed hypothesis followed by premise.
- The remaining design explored the impact of adding different prompts to encourage model correct classification to each sentence and understanding of the current problem.
  - ***"First premise [SEP] Second premise [SEP] Is this statement correct based on previous CTR reports: hypothesis?".***
  - ***"First premise [SEP] Second premise [SEP] Question: Does this imply that: hypothesis?".***
  - ***"Task: Determine Claim Validity \n \n n CTR Report \n First premise [SEP] CTR Report \n Second premise [SEP] Evaluate the Claim: \n hypothesis".***

Also, since the organizers offered the specific lines that contributed to reasoning in a given section presented in both training and validation data, another crucial data-feeding option is whether to feed an entire section for the premise or choose selected lines from a premise section. Some models were trained on the whole section, while others were trained on chosen premise lines.

3.3 Model Selection, Design and training

Based on the following papers results (Wang et al., 2023b; Kanakarajan and Sankarasubbu, 2023; Zhou et al., 2023), experiments were conducted with a variety of different models, including 1) GPT2 (Lagler et al., 2013) 2) Deberta large (He et al., 2020) 3) BioLinkBERT (Yasunaga et al., 2022) 4) Clinical BERT (Alsentzer et al., 2019) 5) Scifive (Phan et al., 2021) 6) BioGPT (Luo et al., 2022).

3.3.1 Model architecture

It is important to modify the model architecture by deciding whether to simply use the last layer and input them into the Fully Connected (FC) layer, or to use the last n-layers from the model and implement average pooling before feeding them to the FC layer, or to direct the output to a convolutional or LSTM layer followed by the FC layer. Experiments were conducted with two alternative options. The first option is to apply mean pooling to the last layer of the model, while the second option is to use GeM pooling on the same layer.

3.3.2 Model Training

BioLinkBERT, Clinical BERT, BioGPT, DeBERTa-large and GPT2 models: Several training approaches have been investigated to improve the generalizability of the model and

its performance. The first approach involves fine-tuning the whole model while using cross-entropy loss. The second approach involves fine-tuning the whole model while using two losses. To improve model performance. The first loss is a cross-entropy loss so penalize the model for wrong prediction; the second loss is contrastive (Chen et al., 2020). The reason behind it is to improve model representation for both classes in the embedding space. The following weights were used: 0.7 for cross-entropy loss and 0.3 for contrastive loss. Following recent practices from the literature, parameter-efficient tuning methodologies as prompt-tuning, LoRA, have been shown to improve model performance over conventional fine-tuning (Fu et al., 2023; Ding et al., 2023). Therefore, the third approach leverages prompt-fine-tuning (Lester et al., 2021) for LM. In prompt-fine-tuning, the data is fed with a prompt to encourage the model to understand the task well, as well as the **"[MASK]"** token. The model task is to predict the correct class in the **"[MASK]"** token. The challenge in prompting lies in the design of the prompt and the model's output. The prompt we used was: ***"First premise [SEP]. Can we infer the hypothesis from the text above? [MASK]".*** The model's output is a binary prediction of either **"yes"** or **"no."**.

Scifive model training: Scifive is based on the T5 (Raffel et al., 2020) generator type, which is an encoder-decoder that transforms all tasks into text-to-text. Instruction fine-tuning has been conducted on the Scifive model with the following template: ***"Determine Claim Validity \n \n. First premise \n \n. Second premise \n\n. Evaluate the following Claim: hypothesis \n \n. Is the assertion accurate? Options: [yes, no]."*** For the loss function of the model BLEU score have been used. The model was constrained to predict either **"valid/invalid"**, or **"correct/incorrect"**, or **"yes/no"**.

3.3.3 Experimental setup

Table 2 shows the hyperparameter setup used during training.

4 Results

In this section, the performance of the proposed models is reported based on the official metric during the dev-phase and test-phase. Error analysis (Lu et al., 2023) was conducted to identify the weaknesses of the proposed models. For the task, the official metric is based on the F1 score and the average faithfulness [2] and consistency [3] scores.

4.1 Dev-phase results

Table 3 shows the results of the developed models on the dev-set, with their training settings. Clearly, the DeBERTa model with different settings showed superior performance compared to other models such as BioGPT, BioLinkBert, ClinicalBERT, GPT2 and Scifive. The Scifive model showed huge performance degradation when compared to BioLinkBert.

The first observation is that changing the pooling technique from mean pooling to GeM pooling, improved model performance by a magnitude of 3%. The second observation is that having two loss function contrastive loss with cross entropy loss improved performance by a magnitude of 3%. The third observation is that building two models for the different cases of reasoning (case single premise, hypothesis, and case of two premises and hypothesis) and including a task description in the data fed improved model performance by 2%. The fourth observation is that prompt-based fine-tuning is better than conventional fine-tuning by magnitude of 1-2%. Another key observation during training is that the model scores a similar f1-score for both classes in most of the settings. The fifth observation is that having data processing as converting numerical quantities to textual representation along with an explanation improves model performance over conventional ones by a magnitude of 1-2%.

4.2 Test-phase results

The results of the proposed system are presented in table 1. Our system ranks in sixth place, with a 0.77 F1-Score, a 0.76 Faithfulness score, and a 0.75 Consistency score. There are correlations between the dev-phase f1-score and the test-phase f1-score, which suggests that a greedy approach to choosing models and their weights is a good approach.

5 Conclusion

The study tested different ways to prepare and load data, as well as more advanced NlM models. It came to the conclusion that the ensemble

[2] Faithfulness measures the extent to which a given system arrives at the correct prediction for the correct reason (Li et al., 2022)

[3] Consistency is a measure of the extent to which a given system produces the same outputs for semantically equivalent problems (Fan et al., 2023)

Combination of selected Models	Leaderboard Results F1-score/ Consistency/Faithfulness	Dev f1-score
BioLinkBert DeBERTa (model 5,8,11,6,10 from table 3)	0.75/0.75/0.79	**0.8945**
DeBERTa (model 8,12,6,9,10 from table 3)	0.743/0.75/0.76	0.8899
DeBERTa (model 8,6,9,11, 10 from table 3)	0.754/0.74/0.75	0.88497
DeBERTa (model 5,8,12,6,9 from table 3)	0.744/0.76/0.80	0.88492
DeBERTa (model 8,6,9,10 from table 3)	**0.765/0.76/0.75**	0.8799
DeBERTa (model 8,11,6,10 from table 3)	0.73/0.75/0.75	0.8749
BioLinkBert DeBERTa (model 8,9,6 from table 3)	0.744/0.75/0.75	0.87474
DeBERTa (model 8,6,9 from table 3)	0.742/0.74/0.78	0.8746

Table 1: Performance of the submitted models on the leaderboard

Hyperparameter	Value
Learning-rate	4e-5 or 5e-6
Schedular	cosine-annealing
Weight decay	1e-3
Epochs	30
Optimizer	Adam
Metric	F1-macro on dev-set

Table 2: The full hyperparameter search space.

model worked well for medical text analysis. DeBERTA, ClinicalBERT, GPT2, Scifive, BioGPT, and BioLinkBert have been investigated and the results show that the DEBERTA model showed better performance compared to other models during the training phase. The final model submitted was an ensemble of various models and techniques. The best-performing model achieved an F1 score of 0.77, a faithfulness score of 0.76, and a consistency score of 0.75, securing the sixth rank in the official leaderboard. Overall, this study has enhanced safe and effective NLMs for complicated medical data analysis in clinical trial reports. Future recommendations could explore other large language models and training techniques, such as LoRA and prefix-tuning, for ingesting medical knowledge into CTRs.

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Hilda Bastian, Paul Glasziou, and Iain Chalmers. 2010. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9):e1000326.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M Zhang. 2023. Large language models for software engineering: Survey and open problems. *arXiv preprint arXiv:2310.03533*.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12799–12807.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

N Holford, SC Ma, and BA Ploeger. 2010. Clinical trial simulation: a review. *Clinical Pharmacology & Therapeutics*, 88(2):166–182.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of*

Model	Training Paradigm	Data Ingestion Prompt	F1-Score
BioGPT	Architecture: Mean Pooling Loss function: Cross Entropy	"premise [SEP] hypothesis".	50
BioLinkBert	Architecture: Mean Pooling Loss function: Cross Entropy	"token_special hypothesis [SEP] token_special premise".	69.7
ClinicalBERT	Architecture: Mean Pooling Loss function: Cross Entropy	"token_special hypothesis [SEP] token_special premise".	63.5
ClinicalBERT	Architecture: Mean Pooling Loss function: Cross Entropy	"premise [SEP] hypothesis".	50
DeBERTa	Architecture: Mean Pooling Loss function: Cross Entropy	"token_special hypothesis [SEP] token_special premise".	80
DeBERTa	Architecture: Mean Pooling Loss function: Cross Entropy	Comparison type [SEP] token_special premise [SEP] premise".	77
DeBERTa	Architecture: Mean Pooling Loss function: Cross Entropy	" premise [SEP] hypothesis".	80
DeBERTa	Architecture: Mean Pooling Loss function: Cross-Entropy and Contrastive Learning	"premise [SEP] hypothesis".	**83**
DeBERTa	Architecture: GeM Pooling Loss function: Cross-Entropy Data preparation: Converted numeric values and abbreviation Two separate models for each comparison type	" premise [SEP] Is this statement correct based on previous CTR reports: hypothesis? ".	82
DeBERTa	Architecture: GeM Pooling Loss function: Cross-Entropy Data preparation: Converted numeric values and abbreviation	" premise [SEP] hypothesis".	82
DeBERTa	Prompting	" premise [SEP] Based on the paragraph above can we conclude that: hypothesis? [MASK] "	81
DeBERTa	Architecture: GeM Pooling Loss function: Cross-Entropy	" premise [SEP] hypothesis".	**83**
GPT-2	Architecture: GeM Pooling Loss function: Cross-Entropy	"premise [SEP] hypothesis".	60
Scifive		"premise [SEP] Question: Does this imply that: hypothesis? ".	50
Scifive		"Task: Determine Claim Validity\n\n CTR Report \n premise [SEP] premise [SEP] f'Evaluate the Claim:\n hypothesis. Options: [correct, incorrect] ".	63.9
Scifive		"Task: Determine Claim Validity\n\n CTR Report \ premise [SEP] f'Evaluate the Claim:\n hypothesis. Options: [valid, invalid] ".	63.73
Scifive		"Determine if a claim is correct based on the following reports.\n Report 1: premise. \n Claim: hypothesis Is the claim correct? \n Options: [yes, no]"	50

Table 3: Models and techniques developed during the experimental and F1-score based on dev-set.

the 18th International Workshop on Semantic Evaluation (SemEval-2024). Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama ai research at semeval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003.

Klemens Lagler, Michael Schindelegger, Johannes Böhm, Hana Krásná, and Tobias Nilsson. 2013. Gpt2: Empirical slant delay model for radio space geodetic techniques. *Geophysical research letters*, 40(6):1069–1073.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Wei Li, Wenhao Wu, Moye Chen, Jiachen Liu, Xinyan Xiao, and Hua Wu. 2022. Faithfulness in natural language generation: A systematic survey of analysis, evaluation and optimization methods. *arXiv preprint arXiv:2203.05227*.

Qingyu Lu, Baopu Qiu, Liang Ding, Liping Xie, and Dacheng Tao. 2023. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt. *arXiv preprint arXiv:2303.13809*.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409.

Benjamin Molinet, Santiago Marro, Elena Cabrio, Serena Villata, and Tobias Mayer. 2022. Acta 2.0: A modular architecture for multi-layer argumentative analysis of clinical trials. In *IJCAI 2022-Thirty-First International Joint Conference on Artificial Intelligence*.

Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Sean R Tunis, Daniel B Stryer, and Carolyn M Clancy. 2003. Practical clinical trials: increasing the value of clinical research for decision making in clinical and health policy. *Jama*, 290(12):1624–1632.

Chaojie Wang, Yishi Xu, Zhong Peng, Chenxi Zhang, Bo Chen, Xinrun Wang, Lei Feng, and Bo An. 2023a. keqing: knowledge-based question answering is a nature chain-of-thought mentor of llm. *arXiv preprint arXiv:2401.00426*.

Weiqi Wang, Baixuan Xu, Tianqing Fang, Lirong Zhang, and Yangqiu Song. 2023b. Knowcomp at semeval-2023 task 7: Fine-tuning pre-trained language models for clinical trial entailment identification. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1–9.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. In *Association for Computational Linguistics (ACL)*.

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. Thifly research at semeval-2023 task 7: A multi-granularity system for ctr-based textual entailment and evidence retrieval. *arXiv preprint arXiv:2306.01245*.

CVCoders at SemEval-2024 Task 4: Unified Multimodal Modelling For Multilingual Propaganda Detection in Memes

Fatemezahra Bakhshande and Mahdieh Naderi and Sauleh Etemadi
Iran University of Science and Technology
{bakhshande.ghazal, mahdieh9816, sauleh}@gmail.com

Abstract

This paper presents our approach to the SemEval 2024 Task 4 on "Multilingual Detection of Persuasion Techniques in Memes." We address the challenge of identifying persuasion techniques in textual and multimodal meme content using a combination of preprocessing techniques and Uni-modal models. Leveraging advanced preprocessing methods, including the OpenAI API for text data, we achieved improved data quality. Our model architecture combines VGG for image feature extraction and GPT-2 for text feature extraction, yielding superior performance. To mitigate class imbalance, we employed Focal Loss as the loss function and AdamW as the optimizer. Experimental results demonstrate the effectiveness of our approach, achieving competitive performance in the task.

1 Introduction

The SemEval 2024 Task 4 [1] focuses on the multilingual detection of persuasion techniques in memes, a crucial endeavor in combating disinformation campaigns prevalent on social media platforms. Memes, being potent vehicles for influencing public opinion, necessitate robust methods for identifying rhetorical and psychological techniques embedded within their textual and visual content. This task spans multiple languages, including Bulgarian, English, and North Macedonian, underscoring the global significance of addressing online misinformation (Dimitrov et al., 2024).

Our system employs a combination of pre-trained models for text and image processing to tackle the challenge posed by Subtask 2b of Task 4. Specifically, we utilize pre-trained language models such as XLM-RoBERTa and GPT-2 for textual feature extraction, while employing VGG and ViT models for image feature extraction. This multimodal approach allows us to effectively capture both textual and visual cues present in memes.

Through our participation in this task, we discovered the importance of advanced preprocessing techniques, particularly in cleaning and standardizing textual data extracted from memes. Leveraging the GPT API for text preprocessing and NLTK for further cleaning proved instrumental in enhancing the quality of our training data. Additionally, we observed the significance of model selection and hyperparameter tuning in achieving competitive performance. Despite encountering challenges in cleaning textual data, our system achieved promising results, demonstrating the efficacy of our approach.

2 Background

The task at hand, Subtask 2b of SemEval-2024 Task 4, revolves around the multilingual detection of persuasion techniques in memes. Memes, which are widely circulated across social media platforms, often contain subtle rhetorical and psychological strategies aimed at influencing public opinion. The goal of the task is to develop models capable of identifying these persuasion techniques embedded within the textual and visual content of memes.

The input to the task consists of textual and visual data extracted from memes in various languages, including Bulgarian, English, and North Macedonian. The textual content of memes may contain linguistic elements such as catchphrases, slogans, or captions, while the visual component typically comprises images or graphics. For example, a meme may feature a humorous image accompanied by a caption containing persuasive language or propaganda.

As for our participation, we focused on Subtask 2b of Task 4, which involves analyzing the presence of persuasion techniques in memes using both

[1] https://propaganda.math.unipd.it/semeval2024task4

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1912–1918

textual and visual features. Our approach combines advanced preprocessing techniques with state-of-the-art models to effectively tackle this challenging task. We draw inspiration from related work in the fields of natural language processing and computer vision, leveraging pre-trained models and techniques to enhance the accuracy and efficiency of our system.

2.1 Related Work

Generative Pre-trained Transformer 2 (GPT-2) is a large language model developed by OpenAI, pre-trained on a dataset of 8 million web pages. It exhibits general-purpose learning capabilities, enabling various tasks such as text translation, question answering, summarization, and text generation (Vincent, 2019; OpenAI, 2019; Piper, 2019).

XLM-R, a large-scale multilingual language model, demonstrates significant performance gains across diverse cross-lingual tasks, outperforming mBERT on tasks such as XNLI and MLQA (Conneau et al., 2020).

Researchers have proposed modified VGG-16 architectures for datasets like CIFAR-10, achieving improved performance with stronger regularization techniques and Batch Normalization (Liu and Deng, 2015).

Vision Transformer (ViT) demonstrates the effectiveness of pure transformer architectures applied directly to image patches for image classification tasks, achieving excellent results compared to convolutional networks (Dosovitskiy et al., 2020).

In recent years, SemEval has incorporated memes into some of its projects, such as Task 6 in 2021[2].

SemEval-2021 Task 6 focused on detecting persuasion techniques in memes, attracting significant participation and highlighting the importance of modeling interactions between text and image modalities (Dimitrov et al., 2021).

SemEval-2023 Task 3[3]. addressed persuasion techniques detection with a multilingual dataset, achieving competitive results using a fine-tuned XLM-RoBERTa large model (Hromadka et al., 2023).

[2] https://propaganda.math.unipd.it/semeval2021task6/

[3] https://propaganda.math.unipd.it/semeval2023task3/

3 System overview

Our system for Subtask 2b of Task 4 in SemEval 2024 employs a combination of algorithms and modeling decisions to detect persuasion techniques in memes based on both textual and visual content. In this section, we outline the key components of our system, including preprocessing steps, model architectures, and training procedures.

3.1 Text Preprocessing

The textual content extracted from memes often contains noise and irrelevant information, which can adversely affect the performance of downstream tasks such as persuasion technique detection. To address these challenges, we employ a series of preprocessing steps to clean and standardize the text data.

3.1.1 OpenAI API for Initial Preprocessing

We utilize the OpenAI API for initial text preprocessing, leveraging its advanced natural language processing capabilities to handle common challenges encountered in meme text extraction. The API effectively identifies and removes extraneous information such as dates, usernames, and additional text that may accompany the original meme content. By leveraging the power of the OpenAI API, we ensure that the text data fed into our system is clean and devoid of irrelevant noise.

3.1.2 Further Cleaning with NLTK

Following the initial preprocessing step, we employ the Natural Language Toolkit (NLTK) for further cleaning and normalization of the text data. The NLTK library provides a wide range of text processing tools, including tokenization, stemming, and stop-word removal, which help standardize the textual content extracted from memes.

3.1.3 Manual Data Correction

Initially, we trained the data without text and solely using images, achieving 45% F1 macro on dev set. So, the major challenge we faced was how to incorporate text.

In instances where the automated text extraction process yielded inaccurate results, manual intervention was necessary to correct these errors. This phase involved a meticulous review of the textual content of problematic memes by human annotators, who then made the necessary adjustments to rectify any discrepancies.

This manual correction process was crucial for ensuring the accuracy and reliability of the textual data used in our system. By meticulously aligning the extracted text with the actual content depicted in the meme images, we mitigated potential biases and inconsistencies that could adversely affect the performance of our system.

In Listing 1, it can be observed that the textual content provided in the 'text' field ("@:\nDer") differs from the text contained within the associated image ('prop_meme_4499.png').

```
{
    "id": "25064",
    "text": "@:\\nDer",
    "image": "prop_meme_4499.png",
    "label": "propagandistic"
}
```

Listing 1: Sample Data Illustrating Textual Content Discrepancy in Manual Data Correction

The actual text present in the meme image is as follows: "Donald Trump Jr. @DonaldTrumpJr.8s\nMuppets have races now? So based on the orange\nI'm guessing Ernie is a Trump and must be \ncancelled immediately!!!\n\nABC News @ABC.7h\nAt only 7 years old, Ji-Young is making history\nas the first Asian American muppet in the\n"Sesame Street" canon. abcn.ws/3FyppJx\n\n'SESAME STREET' DEBUTS\nASIAN AMERICAN MUPPET\nabc NEWS\n\nAP PHOTO/NOREEN NASIR"

3.2 Image Preprocessing

Image preprocessing involves standard techniques such as resizing and normalization to enhance the quality and diversity of the image data. We employ the PyTorch framework for image preprocessing, utilizing built-in functions for resizing and normalization.

In Figure 1, we illustrate the data structure and preprocessing steps employed in our approach.

3.3 Feature Extraction

For textual feature extraction, we fine-tune pre-trained language models, including XLM-RoBERTa and GPT-2, on the meme text data. These models capture semantic and syntactic information embedded in the textual content, enabling effective representation learning for downstream tasks. For image feature extraction, we explore both convolutional neural networks (CNNs) such as VGG and vision transformers (ViTs) to extract visual features from memes. The extracted features

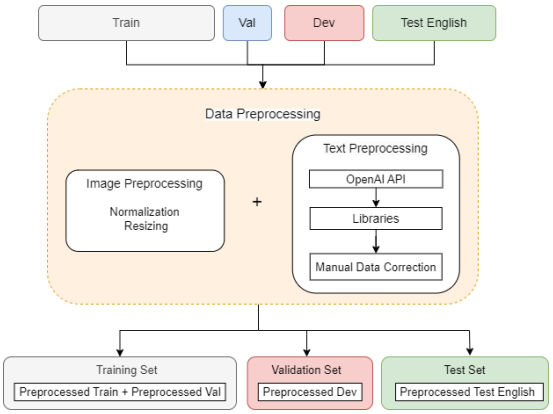


Figure 1: Diagram illustrating the Data structure and preprocessing steps

from both modalities are concatenated to form a multimodal feature representation of the memes.

3.4 Model Architecture

Our model architecture consists of a multimodal fusion layer followed by a classification layer. The multimodal fusion layer combines textual and visual features using concatenation to integrate information from both modalities. The classification layer employs a binary classification approach to predict the presence or absence of persuasion techniques in memes.

In Figure 2, we present the architecture of our Best model, which combines VGG-16 and GPT-2 for Subtask2b.

3.5 Multilingual Considerations

Given the multilingual nature of the task, one initial consideration was how to effectively handle language diversity within the dataset. Initially, we contemplated translating the data into English to leverage state-of-the-art monolingual language models such as BERT. However, inspired by insights from the top-performing submission in last year's TASK 3, (Hromadka et al., 2023) we recognized the efficacy of utilizing pre-trained multilingual models like GPT and XLM-RoBERTa.(Liu et al., 2019) This approach proved advantageous, allowing our system to effectively analyze memes across different languages without the need for explicit translation.

3.6 Overfitting Mitigation Strategies

During the development phase, we encountered challenges related to model overfitting, particularly when using complex architectures such as

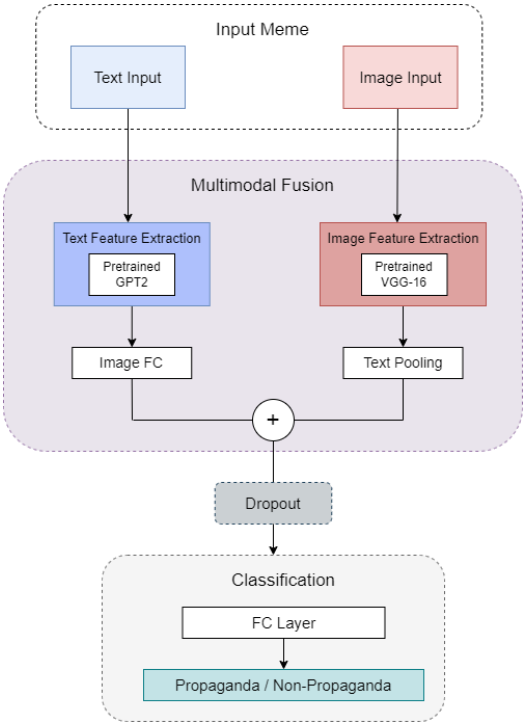


Figure 2: Model architecture combining VGG-16 and GPT-2 for Subtask2b.

a combination of XLM-RoBERTa for text processing and VGG for image analysis. Without proper normalization, our initial model exhibited signs of overfitting, compromising its generalization capabilities. To address this issue, we implemented regularization techniques, including dropout layers, to prevent overfitting and enhance the robustness of our model. These measures proved instrumental in stabilizing the training process and improving the overall performance of our system.

3.7 Training Procedure

We train our model using a combination of supervised learning and fine-tuning techniques. We train the model with the training data merged with the validation data. We employ Focal Loss as the loss function to address class imbalance and AdamW optimizer for gradient descent optimization. Hyperparameters such as learning rate, batch size, and dropout rate are tuned using grid search and cross-validation on the dev set.

3.8 Evaluation and Results

The performance of our system is evaluated using standard evaluation metrics such as macro-F1 score on the test set. We compare our results with baseline models to assess the effectiveness of our approach.

3.9 System Variants

We explore multiple system configurations, including variations in model architectures, preprocessing techniques, and hyperparameter settings. Each variant is evaluated and compared based on its performance on the validation set, allowing us to identify the most effective configuration for the task.

4 Experiment Setup

In this section, we detail the experimental setup used to train and evaluate our system for Subtask 2b of TASK4 2024. This task is a multi-model binary classification task.

4.1 Data Splitting

As shown in Table 1, in this task, we have 1200 samples in the train dataset, 150 samples in the validation dataset, and 300 samples in the `dev_unlabeled` dataset. Initially, the labels for the development dataset were unavailable to be used for testing purposes. However, eventually, these labels were fully accessible under the name of `dev_gold_labels` to the participants, and a dataset consisting of 600 samples was curated to serve as the test dataset, for which the labels have not yet been released.

Data Set/Label	Propagandistic	Non-Propagandistic
Train	800	400
Validation	100	50
Development	200	100

Table 1: Distribution of Datasets

At the beginning of our work, we utilized the same provided training dataset to train our initial model. However, we noticed that the model's accuracy on the training data reached 90% after 2 epochs, but the accuracy on the validation data was not as promising. Despite adjusting hyperparameters, this discrepancy in accuracy did not improve. Therefore, we decided to proceed by using the entire dataset for training our models. This expanded dataset significantly improved the model's performance on the test data.

4.2 Loss Function

After examining the labeled training and validation datasets, we noticed that the data distribution across classes is not uniform. Therefore, we opted to use focal loss alongside binary cross-entropy as the loss function.(Terven et al., 2023)

The formula for Binary Cross-Entropy (BCE) is given by:

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (1)$$

Where:

N is the number of samples,
y_i is the true label of sample i,
$\hat{y}_i$ is the predicted probability of sample i.

The formula for Focal Loss combined with Binary Cross-Entropy (BCE) is given by:

$$FocalLoss + BCE = -\frac{1}{N}\sum_{i=1}^{N}[(1 - \hat{y}_i)^{\gamma} y_i \log(\hat{y}_i) + (1 - y_i)^{\alpha}\hat{y}_i \log(1 - \hat{y}_i)] \quad (2)$$

Where:

N is the number of samples,
y_i is the true label of sample i,
$\hat{y}_i$ is the predicted probability of sample i,
α and γ are hyperparameters controlling the balance and focusing strength

4.3 Hyperparameter Tuning

Hyperparameter tuning played a crucial role in optimizing model performance. We experimented with various hyperparameters, including learning rates, batch sizes, and thresholds, to find the optimal configuration.

Further training parameters are specified in Table 2, in addition to those mentioned above.

Params	Value
number of train epoch	10
train batch size	32
validation batch size	32
weight decay	0.001
learning rate	$1e^{-3}$
threshold	0.39

Table 2: training hyperparameters

Moreover, In Figure 3, the impact of thresholds on Precision, Recall, F-score, and F1-macro metrics is visualized. This analysis provides insights into the trade-offs between these metrics, guiding the selection of an optimal threshold for model evaluation and decision-making.

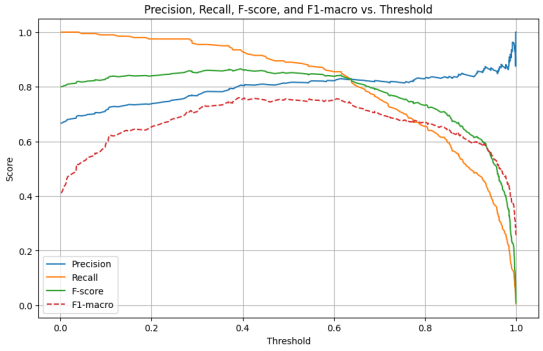


Figure 3: Precision, Recall, F-score, and F1-macro vs. Threshold on the Development Set

4.4 Tools and Libraries

Our system leveraged several external tools and libraries, including:

- OpenAI GPT API[4] for text preprocessing
- PyTorch[5] deep learning framework (v1.9.0) for model implementation
- Hugging Face Transformers[6] library (v4.11.3) for accessing pre-trained language models

4.5 Evaluation Measures

The evaluation of our system's performance was based on macro-F1 score, which accounts for precision and recall across all classes. This metric provides a comprehensive assessment of the model's ability to detect persuasion techniques in memes, considering both true positive and false positive rates (Powers, 2007).

The precision, recall, and F1 score are calculated as follows :

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F\_1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Where:

TP is the total number of true positives,
FP is the total number of false positives,
FN is the total number of false negatives.

[4] https://openai.com/gpt
[5] https://pytorch.org
[6] https://huggingface.co/transformers

F1-macro is calculated as the average of the F1 scores for each class in the classification. It gives equal weight to each class, regardless of its size. The formula for F1-macro is:

$$F1_{\text{macro}} = \frac{1}{N} \sum_{i=1}^{N} F1_i \tag{6}$$

Where:

$$N \text{ is the number of classes,}$$
$$F1_i \text{ is the F1 score for class } i.$$

F1-micro is calculated by considering the total number of true positives, false negatives, and false positives across all classes. It gives equal weight to each instance, regardless of its class. The formula for F1-micro is:

$$F1_{\text{micro}} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{7}$$

Both F1-macro and F1-micro are commonly used to evaluate the performance of classification models, especially in situations where class imbalance exists.(Opitz and Burst, 2019)

5 Results

5.1 Main Quantitative Findings

Our system achieved moderate performance in Subtask 2b of TASK4 2024. On the English test dataset, it attained an F1 macro score of 0.67 and an F1 micro score of 0.74.

The evaluation results of four different model combinations, utilizing the best possible threshold based on the F1 macro on the English Dev dataset, are presented in Table 3. These combinations include VGG + XLM-RoBERTa, VGG + GPT-2, ViT + XLM-RoBERTa, and ViT + GPT-2.

Model	F1 macro	F1 macro Best Threshold
VGG + XLM-RoBERTa	0.58	0.63
VGG + gpt-2	0.71	0.76
ViT + XLM-RoBERTa	0.40	0.53
ViT + gpt-2	0.35	0.51

Table 3: Dev Set Result

Furthermore, the best model, GPT + VGG, was tested on the test data across three languages, and its results are shown in Table 4.

5.2 Quantitative Analysis

To gain deeper insights into our system's performance, we conducted ablation studies and compared different design decisions to identify optimal configurations. We utilized the entire training dataset for these analyses, employing a combination of train, validation, and gold_unlabeled data for training and validation purposes.

Through systematic experimentation, we observed that incorporating focal loss with sigmoid binary activation significantly improved the model's performance, particularly in handling class imbalance issues. Furthermore, training the model using gold_unlabeled data as an additional validation set resulted in notable enhancements in accuracy.

6 Conclusion

This paper presents our approach to the SemEval 2024 Task 4 on "Multilingual Detection of Persuasion Techniques in Memes." Leveraging preprocessing techniques and a multimodal model architecture combining VGG for image features and GPT-2 for text features, our system achieved competitive results on the test dataset in Subtask 2b.

Furthermore, our findings suggest that GPT-2 exhibits greater generalizability than XLM-RoBERTa, with a lower limit on the number of tokens. Insights from our experiments underscore the potential of pre-training models on similar data to enhance performance and generalization.

Looking ahead, future work could focus on pre-training models on meme-specific data and refining preprocessing techniques for extracted text. The large amount of available data presents an opportunity to delve deeper into this aspect, potentially improving model accuracy and robustness in meme analysis tasks.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. arXiv:1911.02116v2.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Language	F1 macro	Baseline F1 macro	F1 micro	Baseline F1 micro
English	**0.67398**	0.25000	**0.74000**	0.33333
Bulgarian	**0.51637**	0.16667	**0.74000**	0.20000
North Macedonian	**0.57653**	0.09091	**0.79000**	0.10000

Table 4: Best Model Result On Test Set

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Semeval-2021 task 6: Detection of persuasion techniques in texts and images. *Proceedings of the thirteenth Workshop on Semantic Evaluation*, Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2021):70–98.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. Unsupervised cross-lingual representation learning at scale.

Timo Hromadka, Timotej Smolen, Tomas Remis, Branislav Pecher, and Ivan Srba. 2023. KInITVeraAI at SemEval-2023 task 3: Simple yet powerful multilingual fine-tuning for persuasion techniques detection. *Proceedings of the Seventeenth Workshop on Semantic Evaluation*.

Shuying Liu and Weihong Deng. 2015. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 1–5, Kuala Lumpur, Malaysia. IEEE.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

OpenAI. 2019. Gpt-2: 1.5b release.

Juri Opitz and Sebastian Burst. 2019. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*. Submitted on 8 Nov 2019 (v1), last revised 8 Feb 2021 (this version, v3).

Kelsey Piper. 2019. A poetry-writing ai has just been unveiled. it's ... pretty good. *Vox*.

David M W Powers. 2007. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, page 37–63.

Juan Terven, Diana M. Cordova-Esparza, and Alfonso Ramirez-Pedraza. 2023. Loss functions and metrics in deep learning. *Journal of Machine Learning Research*.

James Vincent. 2019. Openai has published the text-generating ai it said was too dangerous to share. *The Verge*.

Groningen Team F at SemEval-2024 Task 8: Detecting Machine-Generated Text using Feature-Based Machine Learning Models

Rina Donker and **Björn Overbeek** and **Dennis van Thulden** and **Oscar Zwagers**
Rijksuniversiteit Groningen
{t.r.donker,b.b.j.overbeek,d.l.van.thulden,o.y.zwagers}
@student.rug.nl

Abstract

Large language models (LLMs) have shown remarkable capability of creating fluent responses to a wide variety of user queries. However, this also comes with concerns regarding the spread of misinformation and potential misuse within educational context. In this paper we describe our contribution to SemEval-2024 Task 8 (Wang et al., 2024), a shared task created around detecting machine-generated text. We aim to create several feature-based models that can detect whether a text is machine-generated or human-written. In the end, we obtained an accuracy of 0.74 on the binary human-written vs. machine-generated text classification task (Subtask A monolingual) and an accuracy of 0.61 on the multi-way machine-generated text-classification task (Subtask B). For future work, more features and models could be implemented.

1 Introduction

Recent large language models (LLMs), such as ChatGPT, have shown remarkable capability of creating fluent responses to a wide variety of user queries. This, in combination with increased accessibility to these models, has lead to an increase of machine-generated content over various channels. However, these LLMs come with concerns regarding the potential misuse of such tasks, like spreading misinformation and misuse within the education system (Wang et al., 2023). Therefore, detecting whether a text is human-written or machine-generated is extremely important.

Unfortunately, humans perform only slightly better than chance at this task, as found by Gehrmann et al. (2019). This introduces the need for developing automatic systems that can identify machine-generated texts, in order to mitigate their potential misuse (Wang et al., 2023).

While previous work has been done on identifying machine-generated texts, they either focused on only one or two particular languages, or focused on detecting machine-generated text for a specific LLM or a specific domain (Wang et al., 2023).

For instance, Macko et al. (2023) note that most of the research on machine-generated text detection uses systems that were trained on English datasets and that prior works show that detectors fine-tuned on English data fail to generalize to other languages. This can be seen in the results from Mitchell et al. (2023), where a decrease is seen from 0.946 AUC ROC to 0.537 when working with German data. In addition, Macko et al. (2023) get similar results.

Meanwhile, Sarvazyan et al. (2023) address the issue of detection systems not generalizing across different generation models and domains. They mention that most previous works often overlook that detection systems would be applied to a broad variety of domains, writing styles, and generation models.

Thus, the goal of Task 8 of SemEval 2024 (Wang et al., 2024) is to take a first step into creating a mono- or multilingual system that is able to detect machine-generated text created by different LLMs in different domains. The Task is divided into three subtasks, of which we participate in two. These subtasks are described in Section 2.

In order to tackle this task, we have created multiple feature-based models. The features on which the models have been trained will be described in Section 3. Our decision for creating these feature-based models is supported by our beliefs that a model with carefully crafted features is computationally less expensive than fine-tuning a LLM, while it may be able to achieve equal or better performance because of its ability to generalize (Wang et al., 2023), which is something that LLMs tend to struggle with (Lasri et al., 2022; Wilson et al., 2023). We focused on creating monolingual models for subtask A and B.

Eventually, we created a model that was ranked 95th out of 137 teams for the monolingual track of subtask A with an accuracy of 0.69. For subtask B,

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1919–1925

we ranked 50th out of 77 participating teams with an accuracy score of 0.61.

All of our code for this project can be found on our GitHub repository.[1]

2 Background

The goal of this shared task is to create machine-learning models that are able to detect machine-generated text across multiple languages, generators and domains. The datasets we used to train and test were provided by the task organizers. We took part in two subtasks: A and B.

2.1 Subtask A

The goal of subtask A is to detect whether the text is machine-generated or human-written. Every instance in the dataset contains the text generated by a machine or written by a human and its corresponding label, which is either *human* or *machine*. Besides this, it also includes the model that generated the data and the source of the text.

There is one monolingual dataset, which is made up of English texts only, and one multilingual dataset. We chose to only focus on the monolingual track of this subtask, since we can make language specific feature for this dataset. For the monolingual dataset, the source means the domain of the text, e.g. `reddit` or `wikipedia`. The train set for the monolingual track has 119,757 instances, while the development set contains 5,000 instances.

2.2 Subtask B

The datasets for subtask B are similar to those of subtask A, however, as the goal of this subtask is to detect by which specific text generator the text was created, there are more labels: `human`, `chatGPT`, `cohere`, `davinci`, `bloomz` and `dolly`. The train set consists out of 71,027 instances and the development set includes 3,000 instances. All instances are from sources in English.

3 System Overview

We experimented with four different kind of supervised models and the features that we created ourselves. We will describe each model and feature one by one.

[1] https://github.com/bbjoverbeek/SamEval-2024_Task-8_M4

3.1 Models

Support Vector Machine We chose to implement a Support Vector Machine (SVM; Cortes and Vapnik, 1995), because this has been one of the most fundamental models in statistical machine learning. It has become less popular with the uprising of neural networks, but we expect it will perform well on this task.

Naive Bayes Naive Bayes is a statistical machine learning model that is easy to implement and works well on large datasets. It is based on Bayes' Theorem which calculates the probability of something happening based on previous encounters.

K-Nearest Neighbors Lastly, we implemented the K-Nearest Neighbors algorithm (KNN; Fix and Hodges, 1989). This algorithm is also specialized in classification and is simple to implement.

Feed Forward Neural Network In an attempt to find more complex relationship between different features, we also implement simple neural networks whith several different architectures.

3.2 Features

The following section describes the set of features that we created and experimented with.

Personal pronouns vs. proper pames The first feature focuses on the difference in the usage of personal pronouns, like "he" or "she", versus the usage of proper names, like "Michael" or "Karen" within the text. According to Mitrovic et al. (2023), humans will usually switch to pronouns to refer to a person after using a proper name once or twice in a paragraph, while ChatGPT tends to refer to a person by their proper name more often. For this reason, we thought it would be interesting to see if this effect would be prevalent for other LLMs and if this would have an impact on the results.

Sentence tense This feature focuses on the tense that a sentence is written in. The three tenses we consider are *past*, *present* and *future*. Our goal with this feature was to discover if there are patterns in tense usage that are more commonly used in machine-generated text when compared to human-written text and vice versa.

Sentence voice For all of the sentences, we collect the voice that the sentence was written in. This could be either *passive* or *active*. Similarly to

the sentence tense, we hoped that we could discover patterns that are distinct to either machine-generated text or human-written text, since Mitrovic et al. (2023) mentioned that ChatGPT writes in the passive voice more often than active.

Sentence similarity The sentence similarity calculates how similar a sentence is to its previous and following sentence. In similar fashion to the sentence tense and sentence voice, we want to discover if there are any distinct patterns.

Sentiment We collect the sentiment on sentence level from the text to use as a feature. We believed that there might be a difference between human-written and machine-generated texts in terms of sentiment.

Domain The domain of the text in the train and development data was provided by the task organizers. However, we suspected that this might not be included in the final test set. Therefore, we also included domain as a feature that we could experiment with. We figured that if the domain had a positive influence on the final score, we could build our own classifier that predicts the domain of a text, which can then be used as a feature for our system.

POS-tags and dependency tags Finally, we also included the POS-tags and dependency tags as features. These features contain information about the structure of the text, which we believed could be helpful in distinguishing between machine-generated and human-written text.

4 Experimental Setup

In this following section, we will describe how we created the models and crafted the features.

4.1 Models

Support Vector Machine In order to run the SVM, we made use of the scikit-learn library (version 1.3.2) (Pedregosa et al., 2011).[2] In particular, we used `LinearSVC`. By using the built-in functions of scikit-learn we could train and test the model.

Naive Bayes To build the Naive Bayes classifier, we used the `GaussianNB` classifier from scikit-learn. This type of Naive Bayes classifier assumes that our data is normally distributed.

K-Nearest Neighbors For the KNN algorithm, we used `KNeighborsClassifier` from scikit-learn. When this model is used for subtask A, the number of neighbors is 5. In other cases, the number of neighbors was 15. It is trained and tested in a similar way to the SVM.

Feed Forward Neural Network With the Keras library we created a simple feedforward neural network (FFNN).[3] We experimented with different setups for the neural networks, ranging from 1 up to 4 hidden layers and giving the hidden layers from 8 up to 1024 nodes. The ones that worked best had two or three hidden layers. The size of the layers ranged from 16 up to 256 (depending on the model). All the models use softmax as their activation and Adam for optimization.

4.2 Features

4.2.1 Token-level

Pre-processing With the use of spaCy (version 3.7, using their trained pipeline called *en_core_web_sm*), we could split the full text into tokens.[4] After that, we used the tokens to create our token-level features which will be described below in the following two paragraphs.

Personal pronouns vs. proper names For this feature, we first count how many personal pronouns and proper names the text contains. In order to find the amount of personal pronouns in a text, we use spaCy to find every token that has the POS-tag *pron* (pronoun) and count the number of occurrences.

To collect the number of proper names, we use spaCy's entity recognizer and count every token that has the label *person*.

POS-tags and dependency tags The POS-tags and dependency tags can be easily extracted with spaCy. We use their built-in function to retrieve these tags and then use them as features. For both of these features, we created a bag-of-trigrams.

4.2.2 Sentence-level

Pre-Processing For the sentence-level features, we split the full text into sentences with spaCy. We then use these sentences to extract the features we describe in the remainder of this subsection.

Sentence tense In order to extract the sentence tense, we used spaCy's token-based matching. We

[2] https://scikit-learn.org/

[3] https://keras.io/

[4] https://spacy.io/

created multiple patterns for each sentence tense by using GitHub Copilot[5]. The patterns are made out of combinations of detailed POS-tags, dependency tags, and in some cases, words. The sentences are matched with these patterns and as a result they either get the label *past*, *present* or *future*.

After we collected all the sentence tense labels, we have created trigrams out of these labels and used these in a bag-of-words. We do this by using the `CountVectorizer` that can be found in the skicit-learn library (Pedregosa et al., 2011). The bag-of-trigrams is the feature we use that represents the sentence tense.

Sentence voice Collecting the sentence voice is done in a similar way as the sentence tense. We again use spaCy's token-based matching to determine if a sentence is written in *active* or *passive* voice. The patterns we used were adapted from an example implementation found on Stack Overflow[6]. We then create a bag-of-trigrams in the same way as for the sentence tense and use this as a feature.

Sentence similarity For the sentence similarity feature, the first thing that is done is that each sentence is compared to its previous and following sentence using a sentence-transformers model[7] from Hugging Face (Wolf et al., 2020). We then get two similarity scores per sentence, after which we check for each sentence whether it is most similar to the previous or following one and then give it the value *previous* or *next*, depending on which combination has the highest score. After that, the process is the same as for the sentence tense and voice: we create a bag-of-trigrams to use as a feature.

Sentiment In order to determine the sentiment of a sentence, we used a RoBERTa model from Hugging Face (Liu et al., 2019).[8] Each sentence was assigned one of the following labels: *positive*, *neutral* or *negative*. Afterwards, we again created a bag-of-trigrams so that we could actually use it as a feature.

4.2.3 Document-level

Domain Since domain was already given with the train and test data, we initially used this to experiment with this feature. We trained the models with and without domain to get an insight in the influence of this feature on the final scores. In some cases, the inclusion of domain as a feature improved the score, however, it was only very little. Therefore, we did not find it fruitful to build our own classifier that could predict the domain, especially considering that this classifier may not have been 100% accurate, which would increase the risk of wrong predictions negatively influencing the results.

Evaluation measures In order to evaluate the performance of the model, we calculate precision and recall, the f1-score, and the accuracy. Since the official metric used for subtask A and B is the accuracy, we considered the models with the highest accuracy our best performing models.

We train our different models with every possible combination of features on the training data and evaluate their performance on the development data given by the task organizers. The models with the highest accuracy are the ones we submitted to the shared task.

5 Results

5.1 Subtask A

For subtask A, we handed in three SVMs and three neural networks, since these models received the highest scores on the development set. We achieved the highest accuracy (0.74) on the final test set with a neural network using the sentence tense, sentence voice, sentence similarity and ratio of pronouns and named entities as features. The final ranking released by the task organizers however was not based on the submitted model with the highest score but on the last submitted model, which in our case was the SVM model using the sentence tense, sentence voice and ratio pronouns/named entities as features. This model ranked 95th out of 137 teams. The results of the models we handed in, including our best performing model, can be seen in Table 1. Even though our best performing model obtained an accuracy of 0.74, it is still lower than the RoBERTa baseline, which achieved an accuracy of 0.88.

5.2 Subtask B

For subtask B, we handed in three models, namely one SVM and two neural networks. Our best scoring model was the SVM with an accuracy of 0.61. This model used the sentence tense, sentence voice,

[5] https://github.com/features/copilot
[6] https://stackoverflow.com/a/74594808
[7] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[8] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest

Model	Features	Precision	Recall	F1-score	Accuracy
RoBERTa (baseline)	-	-	-	-	*0.88*
SVM	Tense, Voice	0.73	0.64	0.68	0.69
SVM	Tense, Voice, Ratio PRON/NE	0.73	0.66	0.69	0.69
SVM	Tense, Voice, Similarity, Ratio PRON/NE	0.77	0.64	0.70	0.71
NN 12e-b64-l0.0001	Tense, Voice, Ratio PRON/NE	0.72	0.64	0.68	0.68
NN 8e-b32-l0.0001	Tense, Voice, Similarity, Ratio PRON/NE	0.79	**0.68**	**0.73**	**0.74**
NN 10e-b64-l0.0001	Tense, Voice, Similarity	**0.83**	0.57	0.68	0.72

Table 1: The results of our best models on the monolingual test data of subtask A. The numbers behind the neural networks (NN) stand for the number of epochs (e), the batch size (b) and the learning rate (l) respectively. All the other hyperparameters were the same. The highest scoring model is the neural network with four different features. It is only outscored by another neural network model on precision.

Model	Features	Precision	Recall	F1-score	Accuracy
RoBERTa (baseline)	-	-	-	-	*0.75*
SVM	Tense, Voice, Sentiment, POS DEP, Similarity, Ratio PRON/NE	**0.60**	**0.61**	**0.58**	**0.61**
NN 48e-b32-l0.0005	Tense, Voice, POS, DEP, Ratio PRON/NE	0.54	0.56	0.50	0.56
NN 48e-b32-l0.0005	Tense, Voice, POS, DEP, Similarity, Ratio PRON/NE	0.56	0.56	0.51	0.56

Table 2: The results of our best models on the test data of subtask B provided by the creators of the Shared Task. The numbers behind the neural networks (NN) stand for the number of epochs (e), the batch size (b) and the learning rate (l) respectively. The highest scoring model is the SVM with seven different features.

sentiment, POS-tags, dependency tags, sentence similarity and the pronoun/named entity ratio as features. This model scored the 50th place out of 77 models in total. The results of our best models can be seen in Table 2. Again, our best performing model scores lower than the RoBERTa baseline that has an accuracy of 0.75.

5.3 Discussion

We found that the SVMs and the feedforward neural networks gave the best results on the development set, while the KNN and Naive Bayes algorithms did not perform well. This can be seen in Table 3 in Appendix A, which shows the best results of the different models on subtask A monolingual. Because of these results we decided to focus our attention on feedforward neural networks and SVMs, for both subtask A as well as subtask B.

Our submissions for both subtasks ended up being in the bottom 50% of the total submissions. However, we used older techniques for this task, as we believed that carefully crafting our own features and training these on simpler models would still return good results while being less computationally expensive.

For both subtask A and B we can conclude that a simple FFNN or SVM performs well, but it does not outperform the current state-of-the-art models.

Overall, some features contribute more to the detection of machine-generated text than others. The features that perform well are the sentence tense, sentence voice, sentence similarity and the ratio of personal pronouns and proper names. The tense and the voice of the sentences even appear in all of our best scoring models.

We think that the reason of the effect of ratio of personal pronouns and proper names on the performance is due to the fact that machines tend to use named entities more often than humans, as we described in Section 3.2.

The feature that did not seem effective was sentiment, which only occurs in one of our top models. One of the reasons we think that the sentiment feature did not seem helpful is due to the fact that most of the texts come from sources that are naturally written in neutral tone, such as Wikipedia and arXiv.

6 Conclusion

To conclude, while we still gained quite good results, our models do not outperform the state-of-the-art models. We achieved an accuracy of 0.74 on subtask A and an accuracy of 0.61 on subtask B. Both scores are lower than the RoBERTa baseline.

We discovered that sentence tense, sentence

voice and the ratio between pronouns and named entities seemed to be effective for the classification task, while sentiment did not have that much influence.

In future research, there could be more experimentation with different kinds of machine learning models and more features could be created to further improve the models.

Acknowledgements

This submission has been carried out as a part of the 2023-2024 edition of the master course Shared Task Information Science (LIX026M05) at the University of Groningen. We want to thank Antonio Toral and Lukas Edman for teaching the course. As Antonio Toral was the teacher assigned to our group, we want to thank him especially for supervising our progress, steering us in the right direction and answering all of our questions.

References

C. Cortes and V. Vapnik. 1995. Support vector networks. *Machine Learning*, 20:273–297.

Evelyn Fix and J. L. Hodges. 1989. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Karim Lasri, Alessandro Lenci, and Thierry Poibeau. 2022. Does BERT really agree ? fine-grained analysis of lexical dependence on a syntactic task. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2309–2315, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

Sandra Mitrovic, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. *arXiv:2305.14902*.

Michael Wilson, Jackson Petty, and Robert Frank. 2023. How Abstract Is Linguistic Generalization in Large Language Models? Experiments with Argument Structure. *Transactions of the Association for Computational Linguistics*, 11:1377–1395.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

A Results on Development Set of Subtask A Monolingual

Model	Features	Precision	Recall	F1-score	Accuracy
RoBERTa (baseline)	-	-	-	-	*0.74*
KNN	Tense, Voice, Named Entities	0.637	0.639	0.638	0.637
NB	Tense, Named Entities, Sentiment	0.575	**0.981**	0.725	0.627
SVM	Tense, Voice, Named Entities	0.644	0.888	0.746	0.698
NN 4e-b12-l0.0001	Tense, Voice, Named Entities	**0.687**	0.835	**0.754**	**0.727**

Table 3: This table shows the best performance of each model on the development set of subtask A during our initial experiments. For the NN, we experimented with the number of epochs, batch size and learning rate. We also focused on optimizing the pronouns/named entity feature by using the ratio of pronouns and named entities instead of using absolute values.

Groningen Team A at SemEval-2024 Task 8: Human/Machine Authorship Attribution Using a Combination of Probabilistic and Linguistic Features

Huseyin Alecakir **Puja Chakraborty** **Pontus Henningsson** **Matthijs van Hofslot** **Alon Scheuer**

Faculty of Arts
University of Groningen

Abstract

The emergence of generative language models has put in place the necessity of building models to discern between machine-generated and human-generated text. In this paper, we present our participation in subtasks A and B of the SemEval 2024 Task 8 shared task, which revolves around this problem. Our approach primarily centers on feature-based systems, where a diverse array of features pertinent to the text's linguistic attributes is extracted. Alongside those, we incorporate token-level probabilistic features which are fed into a Bidirectional Long Short-Term Memory (BiLSTM) model. Both resulting feature arrays are concatenated and fed into our final prediction model. Our method under-performed compared to the baseline, despite the fact that previous attempts by others have successfully used linguistic features for the purpose of discerning machine-generated text. We conclude that our examined subset of linguistically motivated features alongside probabilistic features was not able to contribute almost any performance at all to a hybrid classifier of human and machine texts. Our codebase is publicly available on GitHub.[1]

1 Introduction

Large language models capable of generating human-like text have become quite ubiquitous very quickly. There are now many such models which are commonly used to generate text across different domains and in different languages. With their increasing availability and capabilities, it has subsequently become necessary to find ways to distinguish machine-generated text from that which is produced by humans. Humans alone are not able to detect machine-generated text consistently, not even experts in this task (Guo et al., 2023), and current commercial solutions fall short (Chaka, 2023). It is natural then that this problem has seen wide discussion and participation over several domains and languages, including the creation of datasets and proposal of different feature sets and model types (Shamardina et al. 2022; Wang et al. 2024b to name a few), but it is still far from being solved. This leads us to the SemEval-2024 Shared Task 8 that this paper is concerned with "Multidomain, Multimodal and Multilingual Machine-Generated Text Detection" (Wang et al., 2024a). This task is about distinguishing human-written text from machine-generated text in multiple different domains, modalities and across different languages. The languages included in the task are: Arabic, Bulgarian, Chinese, English, Indonesian, Russian and Urdu. The domains are varied and range from Wikipedia pages to arXiv research papers to Reddit posts.

1.1 Research Question

The task of discerning machine-generated texts can be approached using classical feature-based methods or using recent neural methods. The inclusion of multiple domains, languages, and underlying models adds complexity to the problem, but also demands a more universal solution. We therefore find it important not only to strive for high accuracy, but also for explainability and universality based on linguistic concepts. Our research question thus read as the following:

- How well does the linguistically motivated probabilistic model perform for machine-generated text detection and model authorship attribution?

To answer this question, our main strategy uses a combined linear model with document-level features alongside token-level features which have been processed by a BiLSTM, resulting in a method which combines probability-based features with low-level and high-level linguistic features. Our method is inspired by Przybyła et al. (2023), which

[1] https://github.com/rug-1-at-semeval24-task8/code

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1926–1932

used a similar model structure for the AuTexTification shared task (Sarvazyan et al., 2023), achieving results that were close behind an LLM-based model. In our method, however, we employ a linear perceptron instead of a random forest classifier to combine the document-level and processed token-level features, in an attempt to enhance the model's performance and learning. A wide range of features is employed, with our system utilizing stylometric features, entity coherence features, information-theoretic features as well as complementary features such as TF-IDF features for word-level unigrams. An LSTM model proved to attain notably high accuracy in our baseline system, which led to us combining our extracted features with a BiLSTM model. The overview of our system is presented in Figure 1. Our system performs poorly in general and relative to other teams, where we rank at the bottom ten for all tasks that we participated in.

While our primary emphasis remained on feature-based models, we developed a separate model to explore potential performance variations compared to the feature-based approach. In this independent model, we employed a basic LSTM architecture with BERT (Devlin et al., 2018) serving as the embedding layer to acquire sentence embeddings. However, the inclusion of the embedding layer introduced computational overhead, resulting in prolonged processing times. Consequently, we were only able to obtain results for Task B on the test dataset using this architecture.

2 Related Work

Due to the similarities in the architecture and training of different text-generation machines, generated text may possess universal characteristics that distinguish it from text written by humans. Guo et al. (2023) set up a series of human evaluation and linguistics analyses to understand the characteristic features and patterns, where a study by Mitrović et al. (2023) looked at the differences in human vs AI-generated text. The studies found that humans tend to have much more diverse and expressive vocabulary, and often tend to diverge from the topic more than ChatGPT does (Guo et al., 2023; Mitrović et al., 2023). This idea is supported by Gehrmann et al. (2019), who performed a probabilistic analysis of the vocabulary in human- and machine-generated texts and found that generation models tend to have a relatively limited and predictable vocabulary. Some work focuses on stylometric features, as these may be productive in discerning the original author of a text (Li et al., 2014; Pearl and Steyvers, 2012). Wang et al. (2024b) show that models based on such feature sets perform strongly within the domain, but the choice of training dataset may have a notable effect on performance.

Feature-based detectors work fairly well for simple binary classifications in a single domain, but tend to fall short when attempting more complex problems which consist of additional styles and sources of texts (Wang et al., 2024b), where shorter texts can have a negative impact on performance Shamardina et al. (2022). Conversely, language models may prove to be the optimal tool for detecting machine-generated text. Recent attempts mostly use (Ro)BERT(a)-based models (Devlin et al., 2018; Liu et al., 2019) that are pre-trained for language understanding, and fine-tune them using datasets of human- and machine-generated text (Zellers et al., 2019; Shamardina et al., 2022; Guo et al., 2023). These models are then able to detect authorship with varying levels of success. Much of the focus in this area has been on developing useful datasets for fine-tuning and finding optimal models and methods of fine-tuning.

An LSTM, as introduced in Hochreiter and Schmidhuber (1997), is a version of a RNN (recurrent neural network) that utilizes *long term short memory* to deal with issues present in regular RNNs caused by larger gap lengths, which can be especially relevant in NLP tasks such as ours. LSTMs have been used with success to perform authorship attribution (Deibel and Löfflad, 2021; Gupta et al., 2019) which suggests they may be useful in distinguishing human and machine authors as well.

3 Shared Task Set Up

The SemEval-2024 shared task 8 revolved around distinguishing human-written texts and machine-generated texts. It was divided into multiple subtasks. The goal of subtask A was to perform binary classification on a given text to determine whether it is human-written or machine-generated. The monolingual track of this subtask only included text in English, whereas the multilingual track included text in English, Russian, Chinese, Arabic, Urdu, Indonesian and Bulgarian. Subtask B focused on multi-way machine-generated text, where the goal of the task was to determine whether a

given text is written by a human or generated by a machine, and if generated by a machine – which specific language model was it that generated the text?

For all subtasks, we used the datasets provided by the task organizers. These datasets are an extension of the M4 dataset from Wang et al. (2024b). The datasets include texts from multiple domains, such as Reddit discussions, Wikipedia pages and arXiv papers to name a few, as well as multiple languages as stated above. In addition, the dataset for subtask B contains machine-generated texts from multiple models. For more information about the shared task, see Wang et al. (2024a).

4 System Overview

The basic components of our design consist of both document-level and token-level features. Document-level features (detailed in Section 4.1) are extracted directly from the text, and the output features of document-level features are concatenated for further use in an MLP for classification. Token-level features, *i.e.,* the measure of predictability, are the probability of the input text according to a large language model, are fed into a BiLSTM network which converts sequences into a fixed-length representation by concatenating both directions; the details of token-level features are outlined in 4.2. Document-level and token-level features are concatenated and then passed to an MLP for classification. This design remains consistent for both subtask A and subtask B, differing only in the dimensionality of the MLP output representation, which requires adjustments to the number of output classes.

4.1 Document-level features

4.1.1 Perplexity feature

Perplexity serves as a crucial measure of a language model's predictive capability regarding word sequences. Essentially, it gauges the level of surprise a language model experiences when encountering a new sequence of words. A lower perplexity score indicates that the language model excels in predicting the next word in a sequence. It's shown in previous studies that generally the text perplexity generated by large language models (e.g. ChatGPT) is lower than that human written text (Liao et al., 2023). Numerous prior studies have either directly evaluated the efficacy of perplexity in discerning machine-generated data or incorporated it into their models (Liao et al., 2023; Mindner et al., 2023).

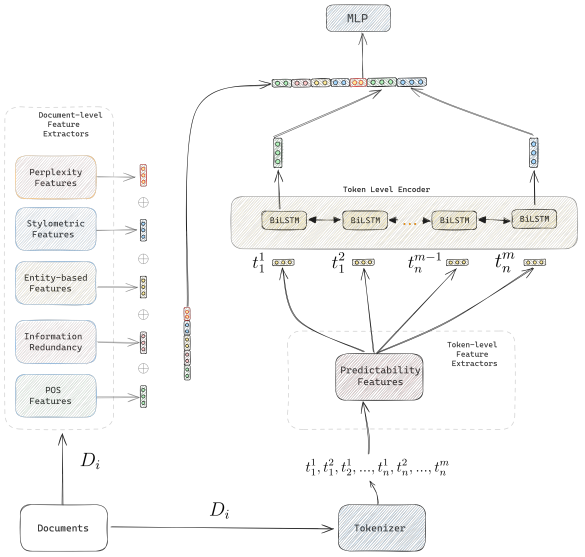


Figure 1: System architecture.

In this study, we employ the XLM-RoBERTa (Conneau et al., 2019) language model to compute the perplexity score for each document. Consequently, each document is represented by 1 perplexity feature.

4.1.2 TF-IDF

We use a text vectorizer to extract term frequency–inverse document frequency (TF-IDF) features based on word-level unigrams. The vocabulary and feature-set for each dataset (A monolingual, A multilingual, B) are calculated separately.

4.1.3 Simple stylometric features

We calculate a small subset of stylometric features. These include: average sentence length by word count; punctuation count, normalized by total number of tokens; number of capitalized words, normalized by total number of words; and the distribution of Part-of-Speech tags in the texts. We make use of the Stanza package (Qi et al., 2020) to perform tokenization, sentence segmentation, and PoS tagging.

4.1.4 Information redundancy

Information redundancy in text may be expressed as lexical or topical repetition. Recent comparisons suggest that machine-generated text is prone to this kind of repetition to some degree (Holtzman et al., 2019), possibly over-repeating words in the output compared to human text (Dou et al., 2021). To calculate information redundancy, we follow the method outlined by Fröhling and Zubiaga (2021).

4.1.5 Entity-based coherence

The inclusion of this feature is based on a hypothesis that human-written text and machine-generated text differ in their use of references to entities throughout the text (Fröhling and Zubiaga, 2021). We extract coherence features using a conventional method which relies on transitions of mention types between sentences (Lapata et al., 2005). An illustration of this process can be found in Figure 2. Due to the limitations of the current co-reference resolution availability, this feature was only used in the monolingual track of subtask A and in subtask B, as these only contained samples in English.

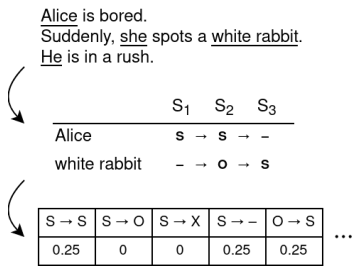


Figure 2: Entity-based coherence illustration.

4.2 Token-level features

4.2.1 Predictability feature

The predictability measurement method, the approach presented by (Przybyła et al., 2023), assesses the likelihood of token sequences using generative language models, distinguishing between machine-generated and human-authored text. Key components of the predictability measurement include:

- *Log-probability of the observed token* t_i^*:

  $\log p(i, t_i^*)$

  This feature measures the likelihood of the observed token given the model's predictions at a specific position in the sequence.

- *Log-probability of the most likely token* w_j *from dictionary* D:

  $\max_{j \in D} \log p(i, w_j)$

  This feature calculates the maximum log probability among all tokens in the model's dictionary, indicating the confidence of the model's top prediction at a particular position.

- *The entropy of the token probability distribution*:

  $-\sum_{j \in D} p(i, w_j) \log p(i, w_j)$

  This feature quantifies the uncertainty of choosing the next token according to the model at a given position.

The XLM-RoBERTa (Conneau et al., 2019) language model is utilized for both Subtasks A and B. Since we only employ the language model, each token is represented by 4 predictability features for all languages, and the maximum sequence length is limited to 128 tokens. The method employs a bidirectional LSTM to distinguish patterns from the sequence of features, without relying on averaging or aggregation functions.

4.3 BERT-LSTM model

Though we mainly focused on feature based system, we have worked on building a simple LSTM model independently as well. For this model, we have used BERT to get sentence embedding, as BERT provide different embedding for the same words based on their context in the sentence. After getting the sentence embedding, we fed it into an LSTM layer, which contains 128 hidden nodes. Subsequently, we have added a linear layer on top of the LSTM layer as the final output layer. The number of output nodes was related to the task it was assigned. For Task B, we have used 6 nodes in the output layer, as there are 6 possible classes. The overview of this system architecture is displayed in Figure 3.

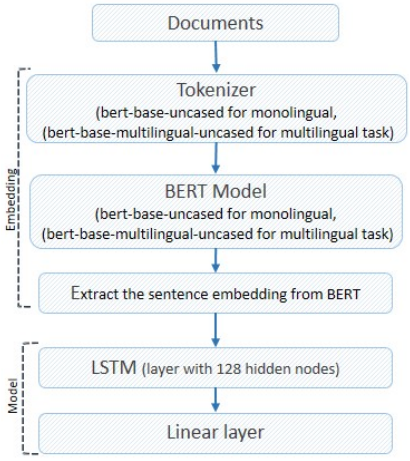


Figure 3: BERT-LSTM system overview.

5 Experimental Setup

We initially divided the training data into two subsets: training and development. We utilized the

development set for testing purposes during the model development phase. However, following the availability of the test data, we incorporated the entire training set for training and the development set for validation. We then evaluated the model's performance on the raw documents in the publicly available test set. For preprocessing, we employed the XLM-RoBERTa tokenizer for features related to perplexity and predictability. Additionally, we utilized the Stanza (Qi et al., 2020) tokenization pipeline for features such as stylometric analysis, entity coherence, information redundancy, and part-of-speech (POS) tagging. Our model was implemented using PyTorch. We employed accuracy as the official evaluation measure.

6 Results

6.1 Feature-based Model

Our numerical results on the test dataset are displayed fully in Table 1. Overall, our system did not perform very well in general or according to official metrics. Our monolingual subtask A model did not learn to differentiate between human and machine texts, and predicted all test examples to be machine-generated. Our subtask B model suffered a similar fate, predicting all test examples to be written by ChatGPT. Our multilingual subtask A was our only model which was able to distinguish between examples to some extent. However, this model also had an extreme bias towards the "machine" label.

Table 1: Overview of our results on each of the subtasks. Values represent accuracy of predictions made on the test set of each subtask.

Subtask	Task Baseline	Our Result	Our Ranking
A Mono.	.884	.525	128/137
A Multi.	.808	.512	61/68
B	.746	.166	74/77

In an attempt to further understand the lack of learning by our models, we examined the raw features produced by our feature extractors on the test set examples. Interestingly, we find that some features did actually differ notably in value for human and machine texts. We calculate means and standard deviations for the raw features on human and machine texts separately, and compare the results using the Cohen's d effect-size metric. Some notable results are shown in Table 3. As expected, the information loss, represented as the norm of the difference between the original document matrix and the truncated matrix, was higher in human texts than in machine texts in the test set, suggesting that the machine texts had more information redundancy, *i.e.,* repetition of information. We observe some interesting findings regarding PoS distribution in the texts, such as higher presence of pronouns, auxiliary verbs, and particles in human texts versus higher presence of nouns, adjectives, and adpositions in machine texts.

Table 2: Confusion matrix for predictions made by our subtask A multilingual model, comparing predicted labels with gold labels.

Subtask A Multi.		Predicted	
		Human	Machine
Gold	Human	406	16259
	Machine	460	17147

Note: Confusion matrices for other subtasks are redundant, as our models only predicted a single label for each of them ('Machine' for subtask A monolingual, and 'ChatGPT' for subtask B).

Table 3: Notable features with effect sizes > 0.3 as calculated on examples from the monolingual A test set. Positive values denote that these features were higher in human texts than in machine texts.

Feature	Effect size (d) (Human − Machine)
Frequency of pronouns	1.59
Frequency of auxiliary verbs	1.49
Frequency of particles	0.8
Frequency of adverbs	0.58
Frequency of verbs	0.54
$\|A - A_{\text{trunc}}\|$ (Information loss)	0.31
$\min(A_{\text{trunc}})$ (Info. redundancy)	-0.51
Frequency of adpositions	-0.75
Punctuation count	-1.03
Frequency of adjectives	-1.31
Frequency of nouns	-1.62

6.2 BERT-LSTM model

We employed BERT (Devlin et al., 2018) to obtain sentence embeddings, a process that significantly increased the computational complexity of our BERT-LSTM system. The model ended up pre-

dicting all the labels on the test set as the same. As a result this system get an 16.67% accuracy on the task B which is not better than a random selection. Due to the time constraint, we could not manage to experiment with Task A.

7 Conclusion

Our overall conclusion is that our examined subset of linguistically motivated features alongside probabilistic features was not able to contribute almost any performance at all to a classifier of human and machine texts. While some features did differ in value between human and machine texts, these differences did not translate into a learning advantage for a hybrid model. Our findings underscore the nuanced challenges inherent in developing robust detection mechanisms for machine-generated text, emphasizing the need for further exploration and refinement of feature engineering strategies to effectively address this evolving domain.

8 Acknowledgments

This submission has been carried out as part of the 2023-2024 edition of the master course Shared Task Information Science (LIX026M05) at the University of Groningen, taught by Lukas Edman and Antonio Toral. We want to humbly express our sincerest gratitude to both Antonio Toral Ruiz and Lukas Edman for their help and support during this task.

We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Hábrók high-performance computing cluster.

Finally, we would like to express our sincere gratitude to the Erasmus Mundus Masters Program in Language and Communication Technologies (LCT) for supporting students who participated in this project.

References

Chaka Chaka. 2023. Detecting ai content in responses generated by chatgpt, youchat, and chatsonic: The case of five ai content detection tools. *Journal of Applied Learning and Teaching*, 6(2).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Robert Deibel and Denise Löfflad. 2021. Style change detection on real-world data using an lstm-powered attribution algorithm. In *CLEF (Working Notes)*, pages 1899–1909.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A Smith, and Yejin Choi. 2021. Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. *arXiv preprint arXiv:2107.01294*.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7:e443.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Shriya TP Gupta, Jajati Keshari Sahoo, and Rajendra Kumar Roul. 2019. Authorship identification using recurrent neural networks. In *Proceedings of the 2019 3rd International Conference on Information System and Data Mining*, pages 133–137.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Mirella Lapata, Regina Barzilay, et al. 2005. Automatic evaluation of text coherence: Models and representations. In *Ijcai*, volume 5, pages 1085–1090.

Jenny S Li, John V Monaco, Li-Chiou Chen, and Charles C Tappert. 2014. Authorship authentication using short messages from social networking sites. In *2014 IEEE 11th International Conference on e-Business Engineering*, pages 314–319. IEEE.

Wenxiong Liao, Zhengliang Liu, Haixing Dai, Shaochen Xu, Zihao Wu, Yiyang Zhang, Xiaoke Huang, Dajiang Zhu, Hongmin Cai, Tianming Liu, et al. 2023. Differentiate chatgpt-generated and human-written medical texts. *arXiv preprint arXiv:2304.11567*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019.

Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. 2023. Classification of human-and ai-generated texts: Investigating features for chatgpt. In *International Conference on Artificial Intelligence in Education Technology*, pages 152–170. Springer.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852*.

Lisa Pearl and Mark Steyvers. 2012. Detecting authorship deception: a supervised machine learning approach using author writeprints. *Literary and linguistic computing*, 27(2):183–196.

Piotr Przybyła, Nicolau Duran-Silva, and Santiago Egea-Gómez. 2023. I've seen things you machines wouldn't believe: Measuring content predictability to identify automatically-generated text. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023). CEUR Workshop Proceedings, CEUR-WS, Jaén, Spain*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains. *arXiv preprint arXiv:2309.11285*.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the the ruatd shared task 2022 on artificial text detection in russian. *arXiv preprint arXiv:2206.01583*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, Malta.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *CoRR*, abs/1905.12616.

SemEval 2024 – Task 10: Emotion Discovery and Reasoning its Flip in Conversation (EDiReF)

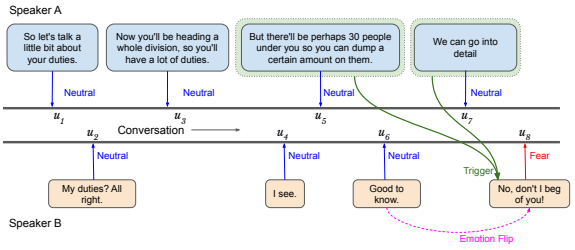
Shivani Kumar[1], **Md Shad Akhtar**[1], **Erik Cambria**[2], **Tanmoy Chakraborty**[3]
[1] IIIT Delhi, India; [2] NTU, Singpore; [3] IIT Delhi, India
{shivaniku, shad.akhtar}@iiitd.ac.in,cambria@ntu.edu.sg
tanchak@iitd.ac.in

Abstract

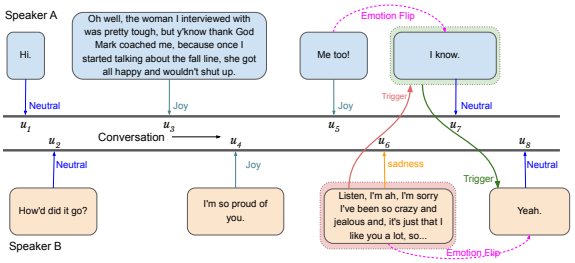
We present SemEval-2024 Task 10, a shared task centred on identifying emotions and finding the rationale behind their flips within monolingual English and Hindi-English code-mixed dialogues. This task comprises three distinct subtasks – emotion recognition in conversation for code-mixed dialogues, emotion flip reasoning for code-mixed dialogues, and emotion flip reasoning for English dialogues. Participating systems were tasked to automatically execute one or more of these subtasks. The datasets for these tasks comprise manually annotated conversations focusing on emotions and triggers for emotion shifts.[1] A total of 84 participants engaged in this task, with the most adept systems attaining F1-scores of 0.70, 0.79, and 0.76 for the respective subtasks. This paper summarises the results and findings from 24 teams alongside their system descriptions.

1 Introduction

In pursuit of one of AI's ultimate objectives, i.e., emulating human behaviour, machines must comprehend human emotions (Ekman, 1992; Picard, 1997). Consequently, Emotion Recognition in Conversation (ERC) has emerged as a vibrant domain within NLP (Hazarika et al., 2018b,a; Zhong et al., 2019; Ghosal et al., 2019; Jiao et al., 2020). The significance of emotion detection amplifies particularly during shifts in the speaker's emotional state. However, merely identifying an emotional transition is insufficient; understanding the catalyst behind the shift is crucial for facilitating informed decisions by other speakers. For instance, identifying the utterance responsible for a customer's transition from a positive emotional state (e.g., joy) to a negative one (e.g., disgust) due to a flawed dialogue system is critical in customer service. Such



(a) Emotion-flip is caused by more than one utterance.



(b) Emotion-flip is caused by the previous utterance. Out of five emotion-flips, we show only two of them ($u_5 \rightarrow u_7$ and $u_6 \rightarrow u_8$) for brevity. Other emotion-flips are $u_1 \rightarrow u_3$, $u_2 \rightarrow u_4$, and $u_4 \rightarrow u_6$ with triggers u_3, u_3, and u_6, respectively.

Figure 1: Examples of emotion-flip reasoning.

insights can serve as feedback to the dialogue system, enabling it to avoid (negative emotion-flip) or replicate (positive emotion-flip) similar utterances to enhance the customer experience in the future.

Emotion-Flip Reasoning (EFR), as outlined by Kumar et al. (2021), presents a novel endeavour aimed at pinpointing the trigger utterances responsible for an emotion-flip within the context of a multi-party conversation. Figure 1 provides illustrative scenarios depicting the essence of EFR. In Figure 1a, Speaker B undergoes a transition in emotion (*neutral→fear*) between utterances u_6 and u_8. Notably, this emotional shift can be attributed to the contributions of Speaker A through utterances u_5 and u_7. The fundamental objective of this task is to discern these trigger utterances (u_5 and u_7) given a target emotion-flip utterance (u_8) and the context preceding it ($u_1 \ldots u_7$).

As an effort to advance research within the do-

[1]The task data is available at https://github.com/LCS2-IIITD/EDiReF-SemEval2024.git.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1933–1946

main of ERC and EFR, this shared task at SemEval 2024 (Ojha et al., 2024) seeks to assess the efficacy of NLP systems in automatically addressing both of these tasks. Furthermore, as technological applications extend beyond English to encompass non-English, multilingual, and code-mixed populations, there is a growing need to broaden the scope of research. To support this cause and further the exploration of code-mixed languages, we advocate for the inclusion of ERC and EFR tasks within Hindi-English (Hinglish) code-mixed conversations. Specifically, the shared task is segmented into three distinct subtasks:

- ***Task A* – ERC in Hindi-English code-mixed conversation:** Given a multiparty code-mixed conversation, tag each utterance with one of the eight emotion labels – *anger*, *disgust*, *fear*, *sadness*, *surprise*, *joy*, *contempt*, and *neutral*.
- ***Task B* – EFR in Hindi-English code-mixed conversation:** Given a multiparty code-mixed conversation along with emotions for each utterance, the goal is to identify the trigger utterance for each emotion-flip in the dialogue.
- ***Task C* – EFR in English conversation:** It is similar to Task B but in monolingual English.

The decision to omit ERC for monolingual English stems from its thorough examination and the abundance of available datasets. Conversely, ERC in Hindi-English code-mixed conversation remains relatively unexplored, and as far as we are aware, no other dataset besides the one outlined in this article is publicly accessible.

Further elaboration on our task data and setting is provided in Sections 3 and Section 4, respectively. The participating teams are outlined in Section 5, with their task outcomes and assessments detailed in Section 6.

2 Related Work

Emotion recognition. Identifying emotions has been a focal point in prior research, with investigations into emotion analysis (Ekman, 1992; Picard, 1997; Cowen and Keltner, 2017; Mencattini et al., 2014; Zhang et al., 2016; Cui et al., 2020) initially centring on standalone inputs devoid of contextual cues. However, recognising the significance of contextual information, the emphasis shifted towards emotion detection within conversations, particularly ERC. Initially, ERC was tackled using heuristic approaches and conventional machine learning techniques (Fitrianie et al., 2003; Chuang and Wu, 2004; Li et al., 2007). However, the recent trend has witnessed a transition towards the adoption of a diverse array of deep learning methodologies (Hazarika et al., 2018a; Zhong et al., 2019; Li et al., 2020; Ghosal et al., 2019; Jiao et al., 2020; Hazarika et al., 2021; Shen et al., 2020; Poria et al., 2017; Jiao et al., 2019; Tu et al., 2022; Yang et al., 2022; Ma et al., 2022).

Emotion and code-mixing. Current studies addressing emotion analysis in code-mixed language primarily centre around isolated social media texts (Sasidhar et al., 2020; Ilyas et al., 2023; Wadhawan and Aggarwal, 2021) and reviews (Suciati and Budi, 2020; Zhu et al., 2022). Despite examinations into aspects like sarcasm (Kumar et al., 2022a,b), humour (Bedi et al., 2023), and offence (Madhu et al., 2023) within code-mixed conversations, the domain of emotion analysis remains largely uncharted, lacking pertinent literature, to the best of our knowledge. Our objective is to address this gap by delving into the under-explored realm of ERC, specifically within Hindi-English code-mixed dialogues in this shared task.

Beyond emotion recognition. The interpretability of emotion recognition within the linguistic domain represents a relatively uncharted avenue of research, with only a limited number of studies delving into this field. Previous works by Lee et al. (2010); Poria et al. (2021); Wang et al. (2023) have focused on investigating the root causes of expressed emotions in text, commonly referred to as 'emotion-cause analysis.' This task involves identifying a specific span within the text that elicits a particular emotion. While on an abstract level, both emotion-cause analysis and emotion-flip reasoning tasks may appear interconnected, they diverge significantly in practice. Emotion-cause analysis aims to pinpoint phrases within the text that provide clues or triggers for the expressed emotion. In contrast, our proposed EFR task pertains to conversational dialogues involving multiple speakers, with the objective of extracting the causes (Kumar et al., 2023a) or triggers behind emotional transitions for a speaker. The triggers comprise one or more utterances from the dialogue history, as illustrated in the two examples in Figure 1.

In this shared task, we tackle the challenge of automatically performing the task of ERC and EFR for code-mixed and monolingual English dialogues in order to further this research direction.

Split	Emotions							Total
	Disgust	Joy	Surprise	Anger	Fear	Neutral	Sadness	
Train	225	1466	1021	911	229	3702	576	8130
Dev	20	156	144	126	39	395	97	977
Test	61	325	238	283	42	943	169	2061

(a) ERC – English

Split	#D with Flip	#U with Flip	#Triggers
Train	834	4001	6740
Dev	95	427	495
Test	232	1002	1152

(b) EFR – English

Split	Emotions								Total
	Disgust	Joy	Surprise	Anger	Fear	Neutral	Sadness	Contempt	
Train	127	1646	444	856	530	4091	572	549	8815
Dev	21	242	68	122	91	652	132	75	1403
Test	21	382	57	150	129	697	167	87	1690

(c) ERC – Hindi

Split	#D with Flip	#U with Flip	#Triggers
Train	344	4406	5565
Dev	47	686	959
Test	58	781	1026

(d) EFR – Hindi

Table 1: Statistics of the English and Hindi datasets for ERC and EFR.

3 Data

English Conversations: We extend MELD (Poria et al., 2019), an established ERC dataset comprising monolingual English dialogues, by incorporating annotations for emotion-flip reasoning. These dialogues are sourced from the popular TV series *F.R.I.E.N.D.S*[2]. Each utterance u is attributed to a specific speaker s and assigned an emotion label $e \in$[*anger*, *disgust*, *fear*, *sadness*, *surprise*, *joy*, *neutral*]. In the context of a speaker's emotional transition, we designate and label trigger utterances as 1 if they induce the speaker's emotional shift – the emotion alters from the speaker's preceding utterance within the same dialogue. In contrast, a label 0 indicates that the utterance bears no responsibility for the emotional transition.

To facilitate the annotation of triggers, we establish a set of guidelines outlined below. Within this framework, a ***trigger*** is defined as any utterance within the contextual history of the target utterance (the utterance for which the trigger is to be identified) meeting the following criteria:

1. An utterance, or part thereof, directly influencing a change in emotion of the target speaker is designated as the trigger.
2. The speaker of the trigger utterance may be different from or the same as the target speaker.
3. The target utterance itself may qualify as a trigger utterance if it contributes to the emotional transition of the target speaker. For instance, if an individual's emotion shifts from *neutral* to *sad* due to conveying a sad message, then the target utterance is deemed responsible for the transition.
4. Multiple triggers may be accountable for a single emotional transition.
5. In cases where the rationale behind an emotional transition is not identifiable from the data, no utterance should be labelled as a trigger.

In total, we have annotated emotion-flip reasoning for $1,161$ monolingual English conversation dialogues, encompassing $8,387$ trigger utterances across $5,430$ emotion-flip instances. Three annotators carefully annotated these dialogues in accordance with the aforementioned guidelines for trigger identification. Among the three, two annotators were male while one was female, all possessing 3-10 years of research experience within the 30-40 age bracket. We calculated the alpha-reliability inter-annotator agreement (Krippendorff, 2011) between each pair of annotators, yielding $\alpha_{AB} = 0.824$, $\alpha_{AC} = 0.804$, and $\alpha_{BC} = 0.820$. By averaging these scores, we derived an overall agreement score of $\alpha = 0.816$. We call the resultant dataset as `MELD-FR`.

Hindi-English Code-mixed Conversations: For code-mixed tasks, we adhere to identical guidelines as those applied to English, selecting code-mixed conversations from a preexisting dialogue dataset called MaSaC (Bedi et al., 2023). The dialogues in the dataset are sourced from the popular Indian TV series '*Sarabhai vs Sarabhai*'[3]. Further, we annotated $11,908$ utterances spanning 449 dialogues, encompassing eight emotion labels (including '*contempt*' alongside the six basic emotions and *neutral*) for the ERC task, achieving a Krippendorff alpha-reliability inter-annotator agreement (Krippendorff, 2011) of 0.85. In the context of EFR, we annotated $7,550$ trigger utterances for $5,873$ emotion-flip occurrences. Mirroring our approach with the English dataset, we engaged experts fluent in both Hindi and English to ensure accuracy. As

[2] https://www.imdb.com/title/tt0108778/

[3] https://www.imdb.com/title/tt1518542/

a measure of quality assurance, the Krippendorff alpha-reliability inter-annotator agreement stands at $\alpha = 0.853$. The resultant dataset is denoted as E-MASAC and EFR-MASAC for the ERC and EFR tasks, respectively. A concise overview of both datasets is presented in Table 1.

4 Task and Background

The idea of the presented shared task is to delve into ERC and EFR within the domain of English and code-mixed dialogues. This section delves into our preliminary investigations for the three subtasks entailed in this collaborative endeavour.

4.1 Shared Task Settings

Task A. In the task of ERC within code-mixed dialogues, participants receive textual utterances as input along with their respective speakers for each dialogue. Their objective is to develop systems capable of autonomously predicting the emotion labels for each utterance. Essentially, the system is presented with a dialogue $D_{erc} = \{(s_1, u_1), (s_2, u_2), ..., (s_n, u_n)\}$, and it must anticipate the emotions e_i for each utterance u_i uttered by speaker s_i. Weighted F-1 score of the emotion classification is used as the evaluation metric for the task of ERC.

Task B. For the code-mixed EFR task, participants receive dialogues along with their corresponding utterances, speakers, and emotions, presented in the format $D_{efr} = \{(s_1, u_1, e_1), (s_2, u_2, e_2), ..., (s_n, u_n, e_n)\}$. Their objective is to anticipate trigger utterances, T, from the context whenever a speaker undergoes an emotion flip. In other words, $T \in \{u_i, ..., u_j\}$ if $s_i = s_j$ and $e_i \neq e_j$. The evaluation metric of choice for this task is the F1 score obtained for trigger utterances.

Task C. The input modelling for Task C mirrors that of Task B, as both tasks revolve around EFR. Here, the data comes from the `MELD-FR` dataset and is present in Monolingual English. Just like Task B, the evaluation is conducted based on the F1 score achieved for trigger utterances.

4.2 Pilot Study

Task A. Our preliminary investigation for the task of ERC in code-mixed setting (Kumar et al., 2023b), we integrate commonsense knowledge with the dialogue representation acquired from a backbone architecture designed for dialogue understanding. We leverage the COMET graph (Bosselut et al., 2019) to extract commonsense knowledge, and subsequently employ context-aware attention (Yang et al., 2019) to integrate this information with the dialogue context. This adaptable module, when combined with RoBERTa (Liu et al., 2019), yields a weighted average F1-score of 0.44 in performance.

Task B. For evaluating the feasibility of our second subtask, we employ FastText multilingual word embeddings[4] for the tokens and perform classification using the proposed model for Task C to obtain an F1-score of 0.27 for trigger identification.

Task C. In our initial exploration (Kumar et al., 2021), we explored a memory-network and transformer-based architecture to address each occurrence of emotion-flip. This approach yielded a trigger-F1 score of 0.53. While these findings surpassed various baselines, the overall performance remains inadequate from a practical standpoint, with an error rate of approximately 50%.

5 Participants

A total of 84 participants engaged in the CodaLab competition organised for the shared task[5], with 24 teams (Shaik et al., 2024; V et al., 2024; Liang et al., 2024; Venkatesh et al., 2024; Yenumulapalli et al., 2024; Niță and Păiș, 2024; Moctezuma et al., 2024; Vyas, 2024; Tareh et al., 2024; Garcia et al., 2024; Wan et al., 2024; Abootorabi et al., 2024; Patel et al., 2024; Siino, 2024; Nguyen and Zhang, 2024; Vaidya et al., 2024; Takahashi, 2024; Alexandru et al., 2024; Rajesh et al., 2024; Shanbhag et al., 2024; Creanga and Dinu, 2024; pan et al., 2024) submitting papers describing their systems. Among the submissions, a prevailing trend emerges with the widespread adoption of Large Language Models (LLMs) such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), GPT (Radford et al., 2019), LLaMa (Touvron et al., 2023), and Mistral (Jiang et al., 2023). Techniques such as fine-tuning, instruction tuning, ensembling, and prompting significantly contribute to enhanced performance in the task. Moreover, many approaches utilise machine learning-based methods including linear regression and SVM. Additionally, some studies explore statistical and rule-based methods such as TF-IDF. While LLMs dominate the approaches for both ERC and EFR, machine learning methods

[4] https://fasttext.cc/docs/en/crawl-vectors.html

[5] https://codalab.lisn.upsaclay.fr/competitions/16769

also remain popular among the participants. Furthermore, there appears to be a notable preference among teams for Task A over Tasks B and C, as evidenced by the higher participation in Task A compared to the latter two. An overview of the top-performing models from various teams for ERC is provided in Table 2, while Table 3 presents the systems for EFR. We summarize some of the techniques used by the top performing systems below.

Using LLMs There exists a prevalent preference for LLMs among teams addressing the ERC and EFR tasks, with approximately 18 methods leveraging LLMs for these endeavours. Notably, BERT and its variants emerge as the most favoured models. Some teams explore larger open-source language models like Zephyr (Tunstall et al., 2023) and Mistral, while at least one team delves into closed-source alternatives such as GPT3.5 (Brown et al., 2020). In the realm of ERC, the leading system (refer to Table 4) integrates DistilBERT (Sanh et al., 2020) with classical machine learning techniques to execute emotion classification optimally. Although the authors experiment with BERT, RoBERTa, and GPT-4 (OpenAI et al., 2023), their most effective model combines DistilBERT with classical ML algorithms. They adopt a two-step approach, initially extracting contextual features from dialogues using an LLM, then inputting these features into classical ML algorithms such as random forests, SVM, logistic regression, and Naive Bayes. Notably, DistilBERT outperforms GPT-4, possibly attributed to the latter's extensive parameter count, necessitating substantial data for meaningful learning. However, our Task A dataset (E-MASAC) encompasses only $\sim$ 8500 utterances, limiting the efficacy of larger models. Conversely, lighter models like DistilBERT exhibit superior adaptability with limited data, capturing nuanced patterns effectively. This finding aligns with observations from various teams, including BITS Pilani, where BERT outperforms Llama.

For the task of EFR as well, LLMs appear to be the predominant choice among the teams. However, intriguingly, the most effective model for this task (refer to Table 5) adopts a classical machine learning approach - XGBoost. Further elaboration on this aspect is provided in Section 6.2.

Classical machine learning and deep learning methods Efficiently capturing context information is crucial in modelling conversations. Several teams explored this aspect, utilising Recurrent Neural Networks (RNNs) like LSTMs and GRUs. Specifically, at least three teams have integrated GRU with Transformers to enhance context capture. Conversely, team TECHSSN adopts a simpler approach, employing LSTM with intelligent embedding layers for both ERC and EFR tasks. However, these methods frequently fall short in comparison to utilising pre-trained LLMs, as outlined in Section 6.2.

Team Name	Backbone Architecture	Model Type
AIMA	GPT3.5 + ML	Ensemble
BITS Pilani	BERT	LLM
CLTeam1	RoBERTa & BERT	LLM+Ensemble
FeedForward	Zephyr	LLM
Hidetsune	SpaCy-v3	ML
IASBS	DistilBERT + ML	LLM+ML
IITK	Transformer + GRU	LLM+DL
INGEOTEC	Bag of Words	Statistical
Innovators	SVM	ML
ISDS-NLP	RoBERTa	LLM
MorphingMinds	LR	ML
RACAI	BERT + ML	LLM+ML
SSN_ARMM	TF-IDF	Statistical
SSN_Semeval10	BERT	LLM
TECHSSN	LSTM	DL
TECHSSN1	RoBERTa	LLM
TransMistral	Mistral 7B	LLM
TW-NLP	MBERT	LLM
UCSC NLP	BERT	LLM
UMUTeam	BERT	LLM
VerbaNexAI Lab	Transformer + GRU	LLM+DL
YNU-HPCC	DeBERTa	LLM

Table 2: Summary of the models according to the submitted system descriptions for Task A (ERC).

Team Name	Backbone Architecture	Model Type
FeedForward	Zephyr	LLM
GAVx	GPT3.5	LLM
IASBS	DistilBERT + ML	LLM+ML
IITK	Transformer + GRU	LLM+DL
Innovators	-	Rule Based
LinguisTech	-	NER Model
SSN_ARMM	TF-IDF	Statistical
TECHSSN	LSTM	DL
TW-NLP	XGBoost	ML
UCSC NLP	BERT + GRU	LLM+DL
UMUTeam	BERT	LLM
YNU-HPCC	DeBERTa	LLM

Table 3: Summary of the models according to the submitted system descriptions for Task B and C (EFR).

Rule-based and statistical methods The surge in deep learning's popularity can be attributed to the remarkable advancements in LLMs. This has led to a decline in the usage of traditional rule-based or statistical approaches, despite their potential to perform comparably in certain scenarios

Rank	Team Name	Results
3	IASBS (Tareh et al., 2024)	0.70
5	FeedForward (Shaik et al., 2024)	0.51
6	TW-NLP (Tian et al., 2024)	0.46
7	TECHSSN1 (Yenumulapalli et al., 2024)	0.45
9	IITK (Patel et al., 2024)	0.45
10	UCSC NLP (Wan et al., 2024)	0.45
11	CLTeam1 (Vaidya et al., 2024)	0.44
13	UMUTeam (pan et al., 2024)	0.43
12	ISDS-NLP (Creanga and Dinu, 2024)	0.43
14	BITS Pilani (Venkatesh et al., 2024)	0.42
15	AIMA (Abootorabi et al., 2024)	0.42
16	SSN_Semeval10 (Rajesh et al., 2024)	0.40
17	Hidetsune (Takahashi, 2024)	0.39
18	INGEOTEC (Moctezuma et al., 2024)	0.39
19	SSN_ARMM (S et al., 2024)	0.38
20	TransMistral (Siino, 2024)	0.36
23	TECHSSN (V et al., 2024)	0.34
24	MorphingMinds (Vyas, 2024)	0.33
26	RACAI (Niță and Păiș, 2024)	0.31
27	Innovators (Shanbhag et al., 2024)	0.28
30	VerbaNexAI Lab (Garcia et al., 2024)	0.24
32	YNU-HPCC (Liang et al., 2024)	0.18

Table 4: Results (Weighted F1) for Task A. Rank is as mentioned in CodaLab. Team Name is as mentioned in the corresponding system description.

alongside more intricate machine learning or deep learning methods. It was pleasant to observe numerous teams incorporating such traditional techniques into this shared task. Notably, at least four teams opted for methods like Bag of Words, TF-IDF, NER based, and rule based approaches. While these methods may not excel in the ERC task, they surprisingly demonstrate superiority in the EFR task. This is reasoned in detail in Section 6.2.

6 Results

In this section, we delve into the outcomes achieved by the participating teams in the shared task outlined earlier. Initially, we will examine the results submitted by the 24 teams, which provided detailed descriptions of their systems. Subsequently, we will present the leaderboard, showcasing the performance rankings of all participants.

6.1 Task A: ERC in Hindi-English code-mixed conversation

The results for Task A are compiled in Table 4. Out of the 24 submitted papers, 22 teams explored the code-mixed ERC task, attaining weighted F1 scores spanning from 0.70 to 0.18. Notably, the foremost twelve teams, up to SSN_Semeval10, opted for LLMs as their architectural preference, yielding top performances. Following closely, RNN-based approaches such as LSTM and classical ML methods like SVM emerged as the subsequent choices. A notable observation is the substantial disparity (approximately 37%) in performance between the leading model and the succeeding system.

Rank	Team Name	Results
1	TW-NLP (Tian et al., 2024)	0.79
2	Innovators (Shanbhag et al., 2024)	0.79
2	UCSC NLP (Wan et al., 2024)	0.79
2	GAVx (Nguyen and Zhang, 2024)	0.79
3	FeedForward (Shaik et al., 2024)	0.77
5	IITK (Patel et al., 2024)	0.56
6	UMUTeam (pan et al., 2024)	0.26
7	IASBS (Tareh et al., 2024)	0.12
9	SSN_ARMM (S et al., 2024)	0.11
11	TECHSSN (V et al., 2024)	0.1
21	YNU-HPCC (Liang et al., 2024)	0.01

Table 5: Results for Task B. F1 score for trigger utterances is our metric of choice. Rank is as mentioned in CodaLab. Team Name is as mentioned in the corresponding system description.

Both leading teams relied on LLMs as their primary architectural framework, yet IASBS diverged by integrating classical ML methods. Their innovative two-phase strategy, combining LLMs for contextual representations and ML techniques for classification, evidently yielded significant improvements. Conversely, the subsequent top model utilised LLMs without any ensembling. Team FeedForward, securing fifth place on the CodaLab leaderboard for Task A, implemented instruction-based finetuning and quantized low-rank adaptation alongside novel techniques like sentext-height and enhanced prompting strategies.

Another intriguing observation arises from the marginal discrepancy (approximately 2%) between strategies based on LLMs and those employing classical ML techniques. Team SSN_Semeval10 refined a BERT classifier, achieving a weighted F1-score of 0.40. Conversely, team Hidetsune took a different approach by translating all code-mixed data into English and employing data augmentation to bolster the 'English'-based ERC dataset. Subsequently, they trained a SpaCy-v3[6] classifier, resulting in a weighted F1-score of 0.39.

6.2 Task B – EFR in Hindi-English code-mixed conversation

Table 5 presents the outcomes for Task B, wherein the highest performance attained a trigger F1-score

[6] https://spacy.io/

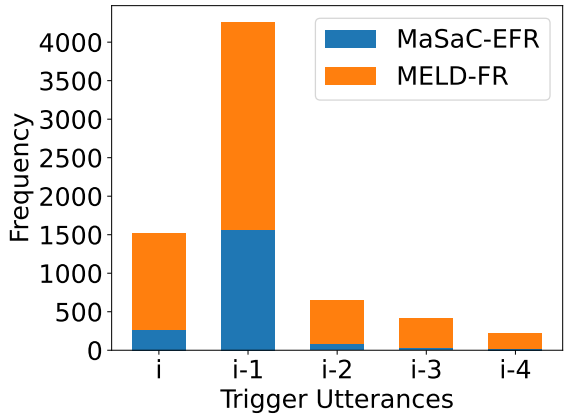


Figure 2: Distribution of triggers for the last four utterances from the trigger utterance i.

Rank	Team Name	Results
2	GAVx (Nguyen and Zhang, 2024)	0.76
3	FeedForward (Shaik et al., 2024)	0.74
5	TW-NLP (Tian et al., 2024)	0.71
7	Innovators (Shanbhag et al., 2024)	0.68
8	UCSC NLP (Wan et al., 2024)	0.68
10	IITK (Patel et al., 2024)	0.6
11	SSN_ARMM (S et al., 2024)	0.26
12	IASBS (Tareh et al., 2024)	0.25
13	TECHSSN (V et al., 2024)	0.24
15	UMUTeam (pan et al., 2024)	0.22
26	YNU-HPCC (Liang et al., 2024)	0.07

Table 6: Results for Task C. F1 score for trigger utterances is our metric of choice. Rank is as mentioned in CodaLab. Team Name is as mentioned in the corresponding system description.

of 0.79. Particularly intriguing is the fact that the leading four teams achieved identical F1-scores, with the top two teams opting for conventional ML and rule-based approaches. This phenomenon stems from the common occurrence wherein a speaker's emotional shift in a conversation at utterance i is predominantly triggered by the $i-1$ utterance. This pattern underscores the significance of the preceding utterance as a trigger. Illustrated in Figure 2 is the trigger distribution within the dialogues of EFR-MASAC and MELD-FR. Evidently, the majority of trigger utterances are the $i-1^{th}$ utterances. Employing XGBoost for trigger classification, the leading team, TW-NLP, secured their position, while the second-ranking team opted for a rule-based approach, designating all $i-1$ utterances as triggers. This strategy led to the attainment of the highest score of 0.79 F1.

6.3 Task C – EFR in English conversation

The outcomes for Task C are displayed in Table 6, revealing the top-performing system achieving an F1 score of 0.76 for the triggers. Impressively, the subsequent results closely trail the best one, exhibiting only a marginal gap of approximately $2\% to 4\%$. Notably, the leading two performers in the task predominantly utilise methods employing LLMs, while the third-best performance is attributed to XGBoost. Illustrated in Figure 2, MELD-FR also grapples with a skewed distribution of trigger utterances, thereby resulting in comparable performances between LLMs and ML-based systems.

6.4 Findings by Participants

Challenge of code-mixing. The dataset utilised in this shared task encompasses Hindi-English code-mixed instances for subtasks A and B, presenting the most formidable challenge of the competition. To address this hurdle, several teams, including TransMistral, FeedForward, and Hidetsune, opted for translation, converting all code-mixed instances into monolingual English before engaging in any classification process. Additionally, teams such as TW-NLP leveraged multilingual LLMs like MBERT to effectively manage code-mixed input.

Effect of data augmentation. Machine learning and deep learning techniques exhibit an insatiable appetite for data, giving rise to circumstances where an abundance of data tends to correlate with improved performance. In light of this conjecture, several teams, including Hidetsune, ventured into experimenting with data augmentation for the ERC task. The general observation revealed an enhancement in performance with the incorporation of more data during model training.

Required context for classification. Emotions are fleeting and are typically influenced by the immediate circumstances in which the speaker finds themselves. As a result, the nearby utterances within a dialogue exert a more pronounced impact on determining the emotional nuances of a speaker compared to utterances further removed in context. This phenomenon is depicted in Figure 2. Consequently, teams such as FeedForward and IITK initially ascertain the requisite extent of context needed for conducting ERC, before proceeding with classification, taking the computed context into consideration.

Challenge of implicit triggers. Emotion flips can generally be attributed to two scenarios: firstly, when something uttered in the dialogue directly prompts the emotion flip, constituting explicit triggers; and secondly, when events external to the

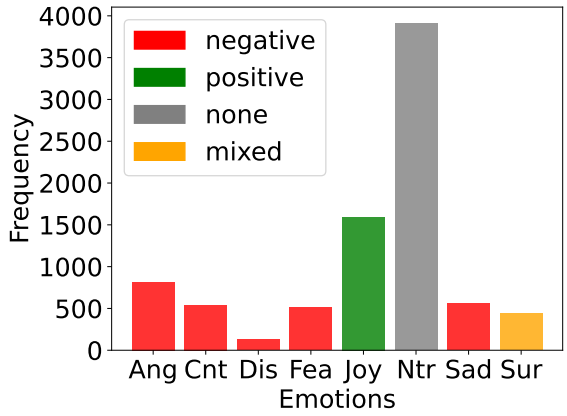


Figure 3: Emotion distribution in E-MASAC. The colors depict the distribution of emotions capturing positive, negative, mixed, and no feelings (Abbreviations: Ang: Anger, Cnt: Contempt, Dis: Disgust, Fea: Fear, Ntr: Neutral, Sad: Sadness, Sur: Surprise).

dialogue, such as an act of theft, occur without explicit mention in the dialogue, representing implicit triggers. In both the EFR-MASAC and MELD-FR datasets, instances of implicit triggers exist where no trigger utterances are marked in the dialogue. These instances present a challenge for the learned models of several teams, including GAVx.

Negative vs positive emotions. The dataset E-MASAC utilises Ekman emotions (Ekman, 1992) as its set of emotion labels, encompassing six emotions and one label for neutral emotions. These emotions include Anger, Contempt, Disgust, Fear, Joy, Neutral, Sadness, and Surprise. Notably, among these emotions, five portray negative feelings (Anger, Contempt, Disgust, Fear, and Sadness), while only one represents positive emotions (Joy). Surprise, on the other hand, can convey either positive or negative emotions depending on the context. Figure 3 displays the distribution of these emotions within E-MASAC. It's evident that as there's only one category for positive emotions, all such instances are classified as joy, leading to a higher frequency of joy compared to other emotions. Moreover, the neutral category has the most instances compared to the others. Consequently, many teams, like IITK, have noted that their models perform better for the neutral and joy labels than for any other emotion.

6.5 Leaderboard

In this paper, we have exclusively examined the outcomes of participants who provided a description of their system(s) for the shared task. The complete array of ranks, team names, and results

Team	Task A	Task B	Task C
MasonTigers	0.78 (1)	0.79 (2)	0.79 (1)
Knowdee	0.73 (2)	0.66 (4)	0.61 (9)
IASBS	0.70 (3)	0.12 (7)	0.25 (12)
-	0.66 (4)	0.07 (20)	0.04 (28)
FeedForward	0.51 (5)	0.77 (3)	0.74 (3)
TW-NLP	0.45 (6)	0.79 (1)	0.71 (5)
TechSSN1	0.45 (7)	0.00 (22)	0.00 (29)
-	0.45 (8)	0.79 (2)	0.68 (8)
IITK	0.45 (9)	0.56 (5)	0.60 (10)
-	0.45 (10)	0.10 (11)	0.15 (23)
CLTeam1	0.44 (11)	0.10 (11)	0.24 (13)
UniBucNLP	0.43 (12)	0.10 (14)	0.06 (27)
UMUTeam	0.43 (13)	0.26 (6)	0.22 (15)
BITS Pilani	0.42 (14)	0.10 (11)	0.24 (13)
AIMA	0.42 (15)	0.10 (12)	0.21 (21)
SSN_Semeval10	0.40 (16)	0.00 (22)	0.00 (29)
OZemi	0.39 (17)	0.00 (22)	0.00 (29)
INGEOTEC	0.39 (18)	0.00 (22)	0.00 (29)
-	0.38 (19)	0.11 (9)	0.26 (11)
-	0.37 (20)	0.07 (19)	0.07 (25)
CUET_NLP	0.37 (21)	0.00 (22)	0.00 (29)
-	0.36 (22)	0.10 (10)	0.22 (19)
TechSSN	0.34 (23)	0.10 (11)	0.24 (13)
MorphingMinds	0.33 (24)	0.10 (16)	0.22 (18)
Z-AGI Labs	0.31 (25)	0.00 (22)	0.00 (29)
RACAI	0.31 (26)	0.10 (11)	0.24 (13)
Innovators	0.28 (27)	0.79 (2)	0.68 (7)
-	0.27 (28)	0.10 (13)	0.21 (20)
-	0.26 (29)	0.10 (15)	0.16 (22)
-	0.24 (30)	0.00 (22)	0.74 (4)
LinguisTech	0.24 (30)	0.00 (22)	0.70 (6)
Team + 1	0.24 (30)	0.09 (17)	0.22 (16)
PartOfGlitch	0.24 (30)	0.00 (22)	0.10 (24)
VerbNexAI Lab	0.24 (30)	0.00 (22)	0.00 (29)
-	0.24 (30)	0.00 (22)	0.00 (29)
silp_nlp	0.24 (31)	0.11 (8)	0.23 (14)
-	0.18 (32)	0.01 (21)	0.07 (26)
-	0.14 (33)	0.09 (18)	0.22 (17)
GAVx	0.08 (34)	0.79 (2)	0.76 (2)

Table 7: Leaderboard from CodaLab. Rank for each task is mentioned in parenthesis. Top three systems are highlighted in green, yellow, and orange.

of the participants in the CodaLab competition, encompassing users who didn't submit a system description, is depicted in Table 7. Although not all teams attempted all three tasks, however, they obtained some score in the leaderboard since they submitted some values for the output. A glimpse at the leaderboard reveals that the best performance for tasks A, B, and C stood at 0.78, 0.79, and 0.79, respectively. One, four, and one team(s) achieved best performance of the three tasks.

7 Conclusion

This paper outlines SemEval 2024 Task 10, covering its goals, data, participants, and results. It

includes three subtasks: emotion identification in code-mixed dialogues and pinpointing triggers for emotion shifts in code-mixed and English dialogues. 84 participants competed on CodaLab, with 24 teams submitting system description papers. Top systems for Tasks A and C used LLM-based architectures, while Task B favored standard ML techniques. Leading systems achieved F1 scores of 0.70, 0.79, and 0.76 across subtasks, indicating impressive performance but also highlighting ongoing challenges for future research.

References

Mohammad Mahdi Abootorabi, Nona Ghazizadeh, Seyed Arshan Dalili, Alireza Ghahramani Kure, Mahshid Dehghani, and Ehsaneddin Asgari. 2024. Aima at semeval-2024 task 10: History-based emotion recognition in hindi-english code-mixed conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1715–1721, Mexico City, Mexico. Association for Computational Linguistics.

Mihaela Alexandru, Călina Georgiana Ciocoiu, Ioana Măniga, Octavian Ungureanu, Daniela Gîfu, and Diana Trandăbăț. 2024. Linguistech at semeval-2024 task 10: Emotion discovery and reasoning its flip in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 399–406, Mexico City, Mexico. Association for Computational Linguistics.

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Trans. Affect. Comput.*, 14(2):1363–1375.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ze-Jing Chuang and Chung-Hsien Wu. 2004. Multi-modal emotion recognition from speech and text. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 9, Number 2, August 2004: Special Issue on New Trends of Speech and Language Processing*, pages 45–62.

Alan S Cowen and Dacher Keltner. 2017. Self-report captures 27 distinct categories of emotion bridged by continuous gradients. *PNAS*, 114(38):E7900–E7909.

Claudiu Creanga and Liviu P. Dinu. 2024. Isds-nlp at semeval-2024 task 10: Transformer based neural networks for emotion recognition in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 636–641, Mexico City, Mexico. Association for Computational Linguistics.

Heng Cui, Aiping Liu, Xu Zhang, Xiang Chen, Kongqiao Wang, and Xun Chen. 2020. Eeg-based emotion recognition using an end-to-end regional-asymmetric convolutional neural network. *Knowledge-Based Systems*, 205:106243.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Siska Fitrianie, Pascal Wiggers, and Leon JM Rothkrantz. 2003. A multi-modal eliza using natural language processing and emotion recognition. In *Text, Speech and Dialogue: 6th International Conference, TSD 2003, České Budéjovice, Czech Republic, September 8-12, 2003. Proceedings 6*, pages 394–399. Springer.

Santiago Garcia, Elizabeth Martinez, Juan Cuadrado, Juan Carlos Martinez-Santos, and Edwin Puertas. 2024. Verbanexai lab at semeval-2024 task 10: Emotion recognition and reasoning in mixed-coded conversations based on an nrc vad approach. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1322–1328, Mexico City, Mexico. Association for Computational Linguistics.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 154–164, Hong Kong, China.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018a. ICON: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604, Brussels, Belgium.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018b. Conversational memory network for emotion recognition in dyadic dialogue videos. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2122–2132, New Orleans, Louisiana.

Devamanyu Hazarika, Soujanya Poria, Roger Zimmermann, and Rada Mihalcea. 2021. Conversational transfer learning for emotion recognition. *Information Fusion*, 65:1–12.

Abdullah Ilyas, Khurram Shahzad, and Muhammad Kamran Malik. 2023. Emotion detection in code-mixed roman urdu - english text. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(2).

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Wenxiang Jiao, Michael Lyu, and Irwin King. 2020. Real-time emotion recognition via attention gated hierarchical memory network. In *AAAI*, volume 34, pages 8002–8009.

Wenxiang Jiao, Haiqin Yang, Irwin King, and Michael R Lyu. 2019. Higru: Hierarchical gated recurrent units for utterance-level emotion recognition. *arXiv preprint arXiv:1904.04446*.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Shivani Kumar, Shubham Dudeja, Md Shad Akhtar, and Tanmoy Chakraborty. 2023a. Emotion flip reasoning in multiparty conversations. *IEEE Transactions on Artificial Intelligence*, pages 1–10.

Shivani Kumar, Atharva Kulkarni, Md Shad Akhtar, and Tanmoy Chakraborty. 2022a. When did you become so smart, oh wise one?! sarcasm explanation in multi-modal multi-party dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5956–5968, Dublin, Ireland. Association for Computational Linguistics.

Shivani Kumar, Ishani Mondal, Md Shad Akhtar, and Tanmoy Chakraborty. 2022b. Explaining (sarcastic) utterances to enhance affect understanding in multimodal dialogues.

Shivani Kumar, Ramaneswaran S, Md Akhtar, and Tanmoy Chakraborty. 2023b. From multilingual complexity to emotional clarity: Leveraging commonsense to unveil emotions in code-mixed dialogues. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9638–9652, Singapore. Association for Computational Linguistics.

Shivani Kumar, Anubhav Shrimal, Md Shad Akhtar, and Tanmoy Chakraborty. 2021. Discovering emotion and reasoning its flip in multi-party conversations using masked memory network and transformer. *arXiv 2103.12360 (cs.CL)*.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 45–53, Los Angeles, CA.

Haifang Li, Na Pang, Shangbo Guo, and Heping Wang. 2007. Research on textual emotion recognition incorporating personality factor. In *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2222–2227. IEEE.

Wei Li, Wei Shao, Shaoxiong Ji, and Erik Cambria. 2020. Bieru: bidirectional emotional recurrent unit for conversational sentiment analysis. *arXiv preprint arXiv:2006.00492*.

Chenyi Liang, Jin Wang, and Xuejie Zhang. 2024. Ynu-hpcc at semeval-2024 task10: Pre-trained language model for emotion discovery and reasoning its flip in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 764–771, Mexico City, Mexico. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hui Ma, Jian Wang, Hongfei Lin, Xuejun Pan, Yijia Zhang, and Zhihao Yang. 2022. A multi-view network for real-time emotion recognition in conversations. *Knowledge-Based Systems*, 236:107751.

Hiren Madhu, Shrey Satapara, Sandip Modha, Thomas Mandl, and Prasenjit Majumder. 2023. Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments. *Expert Systems with Applications*, 215:119342.

Arianna Mencattini, Eugenio Martinelli, Giovanni Costantini, Massimiliano Todisco, Barbara Basile, Marco Bozzali, and Corrado Di Natale. 2014. Speech emotion recognition using amplitude modulation parameters and a combined feature selection procedure. *Knowledge-Based Systems*, 63:68–81.

Daniela Moctezuma, Eric Tellez, Jose Ortiz Bejar, and Mireya Paredes. 2024. Ingeotec at semeval-2024 task 10: Bag of words classifiers. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1104–1109, Mexico City, Mexico. Association for Computational Linguistics.

Vy Nguyen and Xiuzhen Zhang. 2024. Gavx at semeval-2024 task 10: Emotion flip reasoning via stacked instruction finetuning of llms. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 319–329, Mexico City, Mexico. Association for Computational Linguistics.

Sara Niță and Vasile Păiș. 2024. Racai at semeval-2024 task 10: Combining algorithms for code-mixed emotion recognition in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1021–1026, Mexico City, Mexico. Association for Computational Linguistics.

Atul Kr. Ojha, Harish Tayyar Madabushi, Giovanni Da San Martino, A. Seza Doğruöz, Sara Rosenthal, and Aiala Rosá, editors. 2024. *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics, Mexico City, Mexico.

OpenAI, :, and Josh Achiam et al. 2023. Gpt-4 technical report.

ronghao pan, José Antonio García-Díaz, Diego Roldán, and Rafael Valencia-García. 2024. Umuteam at semeval-2024 task 10: Discovering and reasoning about emotions in conversation using transformers. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 689–695, Mexico City, Mexico. Association for Computational Linguistics.

Shubham Patel, Divyaksh Shukla, and Ashutosh Modi. 2024. Iitk at semeval-2024 task 10: Who is the speaker? improving emotion recognition and flip reasoning in conversations via speaker embeddings. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1823–1832, Mexico City, Mexico. Association for Computational Linguistics.

Rosalind W. Picard. 1997. *Affective Computing*. MIT Press, Cambridge, MA, USA.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *ACL*, pages 873–883.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, et al. 2021. Recognizing emotion cause in conversations. *Cognitive Computation*, 13:1317–1332.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Antony Rajesh, Supriya Abirami A, Aravindan Chandrabose, and Senthil Kumar B. 2024. Ssn_semeval10 at semeval-2024 task 10: Emotion discovery and reasoning its flip in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 540–544, Mexico City, Mexico. Association for Computational Linguistics.

Rohith Arumugam S, Angel Deborah S, Rajalakshmi Sivanaiah, Milton R S, and Mirnalinee ThankaNadar. 2024. Ssn_armm at semeval-2024 task 10: Emotion recognition and trigger detection in code-mixed dialogues. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 716–722, Mexico City, Mexico. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

T Tulasi Sasidhar, Premjith B, and Soman K P. 2020. Emotion detection in hinglish(hindi+english) code-mixed social media text. *Procedia Computer Science*, 171:1346–1352. Third International Conference on Computing and Network Communications (CoCoNet'19).

Zuhair Hasan Shaik, Dhivya Prasanna R, Enduri Jahnavi, Rishi Thippireddy, VAMSI MADHAV P S S, SUNIL SAUMYA, and Shankar Biradar. 2024. Feedforward at semeval-2024 task 10: Trigger and sentext-height enriched emotion analysis in multi-party conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 731–742, Mexico City, Mexico. Association for Computational Linguistics.

Abhay Shanbhag, Suramya Jadhav, Shashank Rathi, Siddhesh Pande, and Dipali Kadam. 2024. Innovators at semeval-2024 task 10: Revolutionizing emotion recognition and flip analysis in code-mixed texts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 621–628, Mexico City, Mexico. Association for Computational Linguistics.

Weizhou Shen, Junqing Chen, Xiaojun Quan, and Zhixian Xie. 2020. Dialogxl: All-in-one xlnet for multi-party conversation emotion recognition. *arXiv preprint arXiv:2012.08695*.

Marco Siino. 2024. Transmistral at semeval-2024 task 10: Using mistral 7b for emotion discovery and reasoning its flip in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 297–303, Mexico City, Mexico. Association for Computational Linguistics.

Andi Suciati and Indra Budi. 2020. Aspect-based sentiment analysis and emotion detection for code-mixed review. *International Journal of Advanced Computer Science and Applications*, 11(9).

Hidetsune Takahashi. 2024. Hidetsune at semeval-2024 task 10: An english based approach to emotion recognition in hindi-english code-mixed conversations using machine learning and machine translation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 367–371, Mexico City,

Mexico. Association for Computational Linguistics.

Mehrzad Tareh, Aydin Mohandesi, and Ebrahim Ansari. 2024. Iasbs at semeval-2024 task 10: Delving into emotion discovery and reasoning in code-mixed conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1219–1228, Mexico City, Mexico. Association for Computational Linguistics.

Wei Tian, PeiYu Ji, Lei Zhang, and Yue Jian. 2024. Tw-nlp at semeval-2024 task10: Emotion recognition and emotion reversal inference in multi-party dialogues. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 304–308, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Geng Tu, Jintao Wen, Cheng Liu, Dazhi Jiang, and Erik Cambria. 2022. Context- and sentiment-aware networks for emotion recognition in conversation. *IEEE Transactions on Artificial Intelligence*, 3(5):699–708.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Ravindran V, Shreejith G, Aashika Jetti, Rajalakshmi Sivanaiah, Angel Deborah S, Mirnalinee ThankaNadar, and Milton R S. 2024. Techssn at semeval-2024 task 10: Lstm-based approach for emotion detection in multilingual code-mixed conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 749–756, Mexico City, Mexico. Association for Computational Linguistics.

Ankit Vaidya, Aditya Gokhale, Arnav Desai, Ishaan Shukla, and Sheetal Sonawane. 2024. Clteam1 at semeval-2024 task 10: Large language model based ensemble for emotion detection in hinglish. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 358–362, Mexico City, Mexico. Association for Computational Linguistics.

Dilip Venkatesh, Pasunti Prasanjith, and Yashvardhan Sharma. 2024. Bits pilani at semeval-2024 task 10: Fine-tuning bert and llama 2 for emotion recognition in conversation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 798–802, Mexico City, Mexico. Association for Computational Linguistics.

Monika Vyas. 2024. Morphingminds at semeval-2024 task 10: Emotion recognition in conversation in hindi-english code-mixed conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1207–1211, Mexico City, Mexico. Association for Computational Linguistics.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in hindi-english code-mixed data: A transformer based approach.

Neng Wan, Steven Au, Esha Ubale, and Decker Krogh. 2024. Ucsc nlp at semeval-2024 task 10: Emotion discovery and reasoning its flip in conversation (ediref). In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1503–1508, Mexico City, Mexico. Association for Computational Linguistics.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Baosong Yang, Jian Li, Derek F. Wong, Lidia S. Chao, Xing Wang, and Zhaopeng Tu. 2019. Context-aware self-attention networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):387–394.

Lin Yang, Yi Shen, Yue Mao, and Longjun Cai. 2022. Hybrid curriculum learning for emotion recognition in conversation. In *Proceedings of*

the AAAI Conference on Artificial Intelligence, volume 36, pages 11595–11603.

Venkatasai Ojus Yenumulapalli, Pooja Premnath, Parthiban Mohankumar, Rajalakshmi Sivanaiah, and Angel Deborah S. 2024. Techssn1 at semeval-2024 task 10: Emotion classification in hindi-english code-mixed dialogue using transformer-based models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 820–825, Mexico City, Mexico. Association for Computational Linguistics.

Li Zhang, Kamlesh Mistry, Siew Chin Neoh, and Chee Peng Lim. 2016. Intelligent facial emotion recognition using moth-firefly optimization. *Knowledge-Based Systems*, 111:248–267.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 165–176, Hong Kong, China.

Xun Zhu, Yinxia Lou, Hongtao Deng, and Donghong Ji. 2022. Leveraging bilingual-view parallel translation for code-switched emotion detection with adversarial dual-channel encoder. *Knowledge-Based Systems*, 235:107436.

SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials

Maël Jullien[1], Marco Valentino[3], André Freitas[1,2,3]
[1]Department of Computer Science, University of Manchester, UK
[2] National Biomarker Centre, CRUK-MI, University of Manchester, UK
[3]Idiap Research Institute, Switzerland
[1]{firstname.surname}@manchester.ac.uk
[3]{firstname.surname}@idiap.ch

Abstract

Large Language Models (LLMs) are at the forefront of NLP achievements but fall short in dealing with shortcut learning, factual inconsistency, and vulnerability to adversarial inputs. These shortcomings are especially critical in medical contexts, where they can misrepresent actual model capabilities. Addressing this, we present SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. Our contributions include the refined NLI4CT-P dataset (i.e. Natural Language Inference for Clinical Trials - Perturbed), designed to challenge LLMs with interventional and causal reasoning tasks, along with a comprehensive evaluation of methods and results for participant submissions. A total of 106 participants registered for the task contributing to over 1200 individual submissions and 25 system overview papers. This initiative aims to advance the robustness and applicability of NLI models in healthcare, ensuring safer and more dependable AI assistance in clinical decision-making. We anticipate that the dataset, models, and outcomes of this task can support future research in the field of biomedical NLI. The dataset[1], competition leaderboard[2], and website[3] are publicly available.

1 Introduction

Large Language Models (LLMs) excel in numerous Natural Language Processing (NLP) tasks, as evidenced by their state-of-the-art achievements (Brown et al., 2020; Chowdhery et al., 2022). Despite these advancements, LLMs are prone to several critical vulnerabilities. These include a tendency towards shortcut learning, which may compromise their learning process and accuracy (Geirhos et al., 2020; Poliak et al., 2018; Tsuchiya, 2018). Additionally, they exhibit factual inconsistencies (Elazar et al., 2021) and are sensitive to changes in word distributions (Miller et al., 2020; Lee et al., 2020), data transformations (Xing et al., 2020; Stolfo et al., 2022; Meadows et al., 2023; Rozanova et al., 2023), and adversarial attacks (Li et al., 2020). These issues are particularly concerning as they may lead to an overestimation of LLMs' capabilities in practical applications, a risk that is notably significant in fields requiring high reliability, such as healthcare (Patel et al., 2008; Recht et al., 2019).

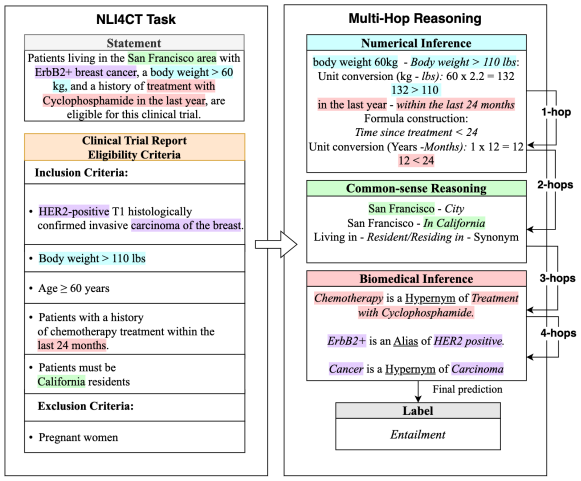


Figure 1: The goal of NLI4CT is to predict the relationship of entailment between a **Statement** and a **CTR** premise (Jullien et al., 2023a). In this task, we introduce a set of perturbations (NLI4CT-P) applied to the statements to test the semantic consistency and faithfulness of NLI models.

Clinical trials play a pivotal role in evaluating the efficacy and safety of novel treatments, thereby significantly contributing to the progress of experimental medicine (Avis et al., 2006). Clinical Trial Reports (CTRs) document the methodologies and outcomes of these trials, serving as a foundation for healthcare professionals to devise and administer experimental therapies. However, the sheer volume

[1]https://github.com/ai-systems/nli4ct
[2]https://codalab.lisn.upsaclay.fr/competitions/16190
[3]https://sites.google.com/view/nli4ct/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1947–1962
June 20-21, 2024

of CTRs, exceeding 400,000 and continually growing (Bastian et al., 2010), renders it impractical for a manual comprehensive analysis of all pertinent literature in treatment planning (DeYoung et al., 2020). In this context, Natural Language Inference (NLI) (Bowman et al., 2015) emerges as a viable solution, facilitating the large-scale interpretation and synthesis of medical evidence. This approach effectively bridges the latest research findings with clinical practice, thereby supporting the delivery of personalized care (Sutton et al., 2020).

Previously, we created the Multi-Evidence Natural Language Inference for Clinical Trial Reports (NLI4CT) dataset, detailed in Jullien et al. (2023a). This dataset, enriched with Clinical Trial Reports (CTRs) and expert-annotated statements for entailment and contradiction, exemplified in Figure 1, served as the foundation for organizing "SemEval-2023 Task 7: Multi-Evidence Natural Language Inference for Clinical Trial Data".

While the preceding version of NLI4CT spurred the creation of models based on Large Language Models (LLMs) (Zhou et al., 2023; Kanakarajan and Sankarasubbu, 2023; Vladika and Matthes, 2023) that demonstrated commendable performance (i.e., F1 score $\approx 85\%$), deploying LLMs in sensitive areas like real-world clinical trials mandates additional scrutiny. This necessitates the invention of new evaluation frameworks that allow thorough behavioural and causal analysis (Wang et al., 2021).

In pursuit of these goals, we present the latest iteration of our dataset, NLI4CT-P, an extension of the original NLI4CT with data perturbations. Moreover, we provide a comprehensive analysis of the systems that participated in "SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials" a task conducted using the NLI4CT-P dataset. This initiative aims to improve our understanding of LLMs behaviour and advance evaluation methodologies for clinical Natural Language Inference (NLI).

The task is structured around the systematic application of controlled interventions, each designed to investigate specific semantic and numerical inference challenges typical of clinical NLI (see Table 1). The interventions enable a comprehensive evaluation of LLMs' reasoning capabilities within a clinical framework, emphasizing robustness, consistency, and faithfulness.

Our efforts aim to significantly contribute to the crafting of more dependable and insightful evaluation standards and metrics for NLI systems, ensuring their reliability and efficacy in healthcare applications.

Original Statement: The primary trial intervention protocol lasts a total of 14 days.
Label: Entailment

Perturbed Statement	Intervention	Type
The primary clinical trial's intervention treatment plan has a duration of 14 days.	Paraphrase	Preserving
The primary clinical trial intervention protocol spans an entire year	Contradiction rephrasing	Altering
Lacks energy refers to whether an individual has/had a lack of energy. The primary trial intervention protocol lasts a total of 14 days	Text appended	Preserving
The primary trial intervention protocol lasts 2 weeks	Numerical paraphrase	Preserving
The primary trial intervention protocol lasts a total of 3 hours	Numerical contradiction	Altering

Table 1: Example of perturbations applied to the statements with the type of intervention and its semantic effect (i.e., preserving vs. altering).

This second iteration is intended to ground NLI4CT in interventional and causal analyses of NLI models (YU et al., 2022). By enriching the original NLI4CT dataset with a novel contrast set derived from targeted interventions to statements in the NLI4CT test and development sets, we establish a direct causal link between these interventions and the anticipated labels. This enhancement introduces two innovative metrics, Consistency and Faithfulness. These metrics allow us to explore specific research objectives with a causal perspective:

- **Consistency:** To examine whether NLI models maintain uniformity in processing semantically equivalent phenomena crucial for inference within clinical NLI contexts.
- **Faithfulness:** To assess the capacity of NLI models to capture and interpret the underlying semantic features required for reasoning over clinical trials, and to change their predictions according to relevant changes of such features.

This paper introduces SemEval-2024 Task 2 – Safe Biomedical Natural Language Inference for Clinical Trials – (NLI4CT-P) presenting a detailed analysis of the performance of the participating systems. We report the following conclusions:

Challenges in Clinical NLI: Despite improvements achieved via the application of Large Language Models (LLMs), Clinical NLI remains a significant challenge. With the highest F1 score achieved in this task being 0.8 (Liu and Thoma, 2024; Guimarães et al., 2024) (FZI-WIM, Lisbon Computational Linguists), leveraging Mixtral-8x7B-Instruct models. This emphasises the necessity for the development of more robust and reliable systems capable of dealing with the challenges of real-world clinical application.

Importance of Faithfulness and Consistency Evaluation: The incorporation of Faithfulness and Consistency metrics into our evaluation framework underscores the unpredictability of current systems and the limitations inherent in relying solely on F1 score for comprehensive analysis.

Superiority of Generative Models: Generative models have been shown to outperform discriminative models in terms of F1 score (+0.025), Faithfulness (+0.15), and Consistency (+0.037).

Value of Additional Data: Leveraging additional training data in the form of instruction tuning or medical NLI datasets has been shown to produce significant performance gains. When augmented with extra data, systems exhibit notable enhancements, recording improvements of +0.056 in F1 score, +0.132 in Faithfulness, and +0.062 in Consistency relative to their counterparts.

Impact of Prompting Strategies: The study highlights that the choice of prompting strategy plays a crucial role in influencing model performance. Specifically, zero-shot prompting has been shown to provide notable enhancements, with an average increase of +0.025 in F1 score, and marginal gains of +0.001 in both Faithfulness and Consistency, compared to the outcomes achieved with few-shot prompting techniques.

Efficacy of Mid-Sized Architectures: Mid-sized architectures, possessing 7B to 70B parameters, offer a cost-effective alternative capable of matching or surpassing larger models in key performance metrics like F1, Faithfulness, and Consistency. Compared to models exceeding 70B parameters, these mid-sized models report a slight improvement of +0.01 in F1 score, albeit with minor reductions of -0.03 in Faithfulness and -0.01 in Consistency. Against models below 7B parameters, however, they show notable enhancements, achieving +0.10 in F1 score, +0.40 in Faithfulness, and +0.19 in Consistency.

2 Task Description

SemEval-2024 Task 2 is a textual entailment task, each instance in NLI4CT contains a CTR premise and a related statement. These premises range from 5 to 500 tokens in length and provide details about a trial's results, eligibility criteria, interventions, or adverse events. Corresponding statements are concise sentences, containing 10 to 35 tokens, that make some claim about the premise information (refer to Table 1 for examples). The task is to classify the inference relation between a CTR premise, and a statement as either entailment or contradiction, exemplified in Figure 1 The dataset features two distinct types of instances: single instances, where a statement discusses a single CTR, and comparison instances, which involve statements that compare and contrast two CTRs.

3 Dataset

The premises used in the NLI4CT dataset (Jullien et al., 2023a) are derived from 1,000 publicly accessible, English-language breast cancer Clinical Trial Reports (CTRs) published on ClinicalTrials.gov a resource managed by the U.S. National Library of Medicine. This dataset complies with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. The original NLI4CT collection includes 2,400 expert-annotated statements, premises and associated labels. These are distributed across training, testing, and development sets in a 70/20/10 ratio.

We have advanced the methodology of the previous NLI4CT dataset by incorporating interventions to create a contrast set, enabling a systematic behavioural and causal analysis of models evaluated in the competition. This enhanced version is referred to as NLI4CT-P (Perturbed). The construction of the contrast set involves four semi-automated, controlled interventions applied to the statements from the NLI4CT test and development set. It's important to note that the specifics of these interventions were kept undisclosed until the completion of the competition's testing phase on January 31st 2024.

3.1 Interventions

We delineate and implement the four interventions in the following manner:

Paraphrasing and Contradiction Rephrasing Clinical texts frequently contain acronyms and aliases, which can hinder the performance of clinical NLI models (Grossman Liu et al., 2021; Jimeno-Yepes et al., 2011; Pesaranghader et al., 2019; Jin et al., 2019). Moreover, these models can fall prey to shortcut learning, where they make inferences based on syntactic patterns rather than semantic understanding (Geirhos et al., 2020). To evaluate this phenomenon, original statements were rephrased using different vocabulary and syntax. Paraphrasing was employed to retain the original meaning and label (row 1 Table 1), while contradiction rephrasing created new statements that directly contradict the original statement, and are therefore always labelled as contradictions (row 2 Table 1).

Numerical Paraphrasing and Contradiction Large Language Models (LLMs) have shown limitations in consistent numerical and quantitative reasoning (Patel et al., 2021; Ravichander et al., 2019; Galashov et al., 2019), an essential aspect for tasks like NLI4CT that demand such inferences. To evaluate the models' capabilities in this area, operands and numerical units within the hypotheses were altered (rows 4 and 5 Table 1). This modification either preserved or inverted the initial entailment label.

Appending Text LLMs are often challenged by complex reasoning when dealing with extended premise-hypothesis pairs (Liu et al., 2021). We test this in a clinical setting by appending biomedical definitions from the NCI Thesaurus to the original statements (row 3 Table 1). The added definitions, ranging from 15 to 20 tokens in length, almost double the average statement token length. Despite the definitions not being independently verifiable against the premises, these definitions are regarded as 'ground truth', they are universally true and remain neutral in relation to the premises. Since they neither assert nor verify any premise-specific information, within the scope of our task, appending such neutral text is categorized as a 'preserving' intervention.

These interventions, other than the text appending, were performed by prompting ChatGPT 3.5 and Whisper APIs (Brockman et al., 2023) with human-in-the-loop correction to address any errors (Gilardi et al., 2023). Each statement in the test and development sets underwent each type of intervention process three times. This did not extend to the training set, as the aim was to prevent models from learning the patterns of intervention. Although attempts were made to apply numerical paraphrasing and contradiction interventions, they were not always feasible. This was due to the absence of numerical data or units in the original statements, and when the quality of the perturbed statements was deemed substandard, they were excluded during the manual review phase. consequently, this resulted in a markedly reduced count of numerically perturbed statements within the dataset. The prompts used to perform the interventions are available in the appendix.

4 Evaluation

SemEval-2024 Task 2 is devised as a binary classification challenge, with the Macro F1-score being utilized to gauge the foundational performance of the participating systems. This evaluation is conducted on the original NLI4CT test set, serving as a control metric, rather than on the NLI4CT-P test set, which contains exclusively perturbed statements. Although the Macro F1 score is instrumental in measuring overall model performance by highlighting precision and recall across various classes, it inherently lacks the capability to fully capture the sophisticated understanding and reasoning skills essential for effective Natural Language Inference (NLI). Specifically, the F1 score does not assess a model's capacity to adjust to subtle semantic shifts or evaluate the resilience of its predictions when faced with interventions that either modify or maintain the semantic integrity of statements. This gap highlights the necessity for more advanced metrics capable of offering deeper insights into a model's interpretative and reasoning proficiency. In response to this need and inspired by recent advancements in causal analysis within the NLP domain (Stolfo et al., 2022), we introduce two novel evaluation metrics aimed at examining the causal effects of interventions on model performance.

Faithfulness gauges the degree to which a system's predictions are both accurate and grounded in the correct rationale. Intuitively, this is estimated by measuring the ability of a model to correctly adjust its predictions when exposed to interventions that modify the meaning (semantic altering) of the statement. Specifically, for a set of N statements x_i in the contrast set (C), alongside their corresponding original statements y_i and the model predictions denoted as $f()$, faithfulness is quantified using the

Set	Original	Appended definition	Paraphrase	Contradiction rephrasing	Numerical paraphrase	Numerical Contradiction	Total
Dev	200	600	600	600	64	78	1942
Test	500	1500	1500	1500	224	276	5000

Table 2: Distribution of statement counts across the sets of NLI4CT-P

formula presented in Equation 1.

$$Faithfulness = \frac{1}{N}\sum_{1}^{N}|f(y_i) - f(x_i)|$$
$$x_i \in C : \text{Label}(x_i) \neq \text{Label}(y_i), \text{ and } f(y_i) = \text{Label}(y_i) \quad (1)$$

Consistency assesses a system's capability to generate identical outcomes for semantically equivalent inputs. This measure evaluates whether a system can uniformly predict the same label for both the original and contrast statements under interventions that do not alter the semantic content (semantic preserving) of the statements. The key aspect here is the uniformity in representing semantic concepts across different statements, irrespective of the correctness of the final prediction. For N statements x_i in the contrast set (C), alongside their original counterparts y_i, and model predictions $f()$, consistency is determined as follows:

$$Consistency = \frac{1}{N}\sum_{1}^{N}1 - |f(y_i) - f(x_i)| \quad (2)$$
$$x_i \in C : \text{Label}(x_i) = \text{Label}(y_i)$$

The Macro F1 score provides a foundational benchmark for basic model performance, serving as a control metric, the core objective of Task 2 is towards enhancing model quality and dependability through systematic causal analysis. The pursuit here is not only for high performance in a traditional sense but for models that demonstrate a more reliable and robust application of natural language, reflecting a more nuanced approach to evaluating system capabilities, and allowing for developing safer, ethical, and trustworthy clinical systems.

5 Results and Discussion

106 participants registered to the SemEval-2024 Task 2 competition contributing over 1200 individual submissions and 25 system overview papers, presented in Table 3. Please note that our analysis focuses exclusively on systems that are detailed in system overview papers and for which official leaderboard results have been provided. Generally, participants tend to submit the highest-scoring results to the leaderboard, regardless of whether the system achieving these results represents the primary contribution of their paper. This approach ensures that our report reflects the peak performance levels achieved, albeit potentially overlooking the main systems of interest described in the papers.

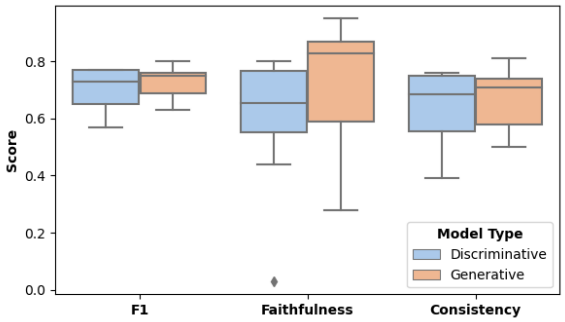


Figure 2: Comparative Analysis of F1, Consistency, and Faithfulness Across Model Types

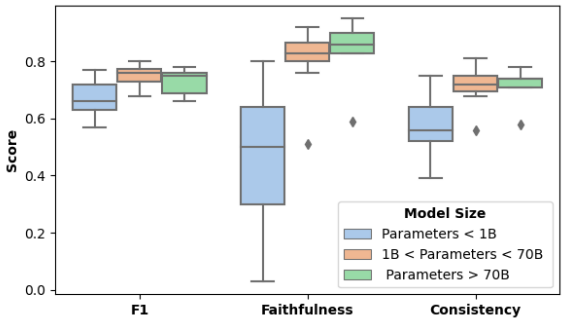


Figure 3: Comparative Analysis of F1, Consistency, and Faithfulness Across Model Parameter Numbers

5.1 Architectures

In the SemEval-2024 Task 2 submissions, a diverse range of 12 different architectures was employed, as detailed in Table 4. The predominant choice among participants was Mistral-based architectures, accounting for 7 out of 25 submissions, closely followed by DeBERTa with 5 out of 25. The majority of submissions utilised generative

Work	F1	F	C	Average Score	Architecture	Inference Strategies	Fine-Tuning	Dataset Augmentation
FZI-WIM (Liu and Thoma, 2024)	**0.8**	0.9	0.73	0.81	Mixtral-8x7B-Instruct	CoT	Yes	GPT-4, bart-large-mnli Instruction Dataset
Lisbon Computational Linguists (Guimarães et al., 2024)	**0.8**	0.83	0.72	0.78	Mistral-7B-Instruct-v0.2	Zero-shot	Yes	Mistral-7B-Instruct-v0.2 dataset expansion
NYCU-NLP (Lee et al., 2024)	0.78	0.92	**0.81**	**0.84**	SOLAR (10.7B)	Zero-shot	Yes	OpenChat v3.5, Intervention Reduction
Edinburgh Clinical NLP (Gema et al., 2024)	0.78	**0.95**	0.78	**0.84**	GPT-4	Zero-shot	No	-
YNU-HPCC (Zhang et al., 2024)	0.77	0.67	0.73	0.72	DeBERTa-v3-large	Discriminative	Yes	MultiNLI, FeverNLI, ANLI, LingNLI, WANLI, Back Translation
BD-NLP (Nath and Samin, 2024)	0.77	0.79	0.76	0.77	DeBERTa-lg	Discriminative	Yes	-
CaresAI (Abdel-Salam et al., 2024)	0.77	0.76	0.75	0.76	Ensemble of DeBERTas	Discriminative	Yes	-
TüDuo (Smilga and Alabiad, 2024)	0.76	0.84	0.75	0.78	Flan-T5 XL	Few-shot	Yes	GPT-3.5-Turbo Instruction Dataset
RGAT (Chakraborty, 2024)	0.76	0.86	0.74	0.79	GPT-4	Zero-shot	No	-
DFKI-NLP (Verma and Raithel, 2024)	0.75	0.81	0.68	0.75	Mistral 7B	Zero-shot	Yes	Meta-Inventory dataset expansion, MedNLI
D-NLP (ALTINOK, 2024)	0.75	0.83	0.74	0.77	Gemini Pro	Zero-shot	No	-
LMU-BioNLP (Sun et al., 2024)	0.75	0.86	0.69	0.77	Mistral-7b	Zero-shot	Yes	GPT-3.5, GPT4 dataset expansion, and instruction tuning dataset
DKE-Research (Wang et al., 2024)	0.74	0.8	0.75	0.76	DeBERTa-l	Discriminative	Yes	GPT-3.5, TF-IDF dataset expansion
Puer (Dao et al., 2024)	0.72	0.59	0.64	0.65	Biolinkbert-large	Discriminative	Yes	-
UniBuc (Micluța-Câmpeanu et al., 2024)	0.71	0.83	0.72	0.75	SOLAR 10B	few-shot	No	-
iML (Akkasi et al., 2024)	0.7	0.28	0.52	0.50	SciFive	Zero-shot	Yes	-
CRCL (Brutti-Mairesse, 2024)	0.7	0.87	0.7	0.76	Mixtral-8x7B	CoT, OPRO optimization	No	-
IITK (Mandal and Modi, 2024)	0.69	0.9	0.71	0.77	Gemini Pro	Zero-shot, ToT and CoT	No	-
0x.Yuan (Lu and Kao, 2024)	0.68	0.51	0.56	0.58	Mixtral-8x7B	multi-agent debating framework	No	-
Saama Technologies (Kim et al., 2024)	0.66	0.59	0.58	0.61	Gemini Pro, mistral-7B-instruct-v0.2	CoT, Few-Shot	Yes	-
TLDR (Das et al., 2024)	0.66	0.5	0.58	0.58	SciFive-base, DeBERTa-v3-base	Zero-shot	No	-
Concordia University (Marks et al., 2024)	0.66	0.03	0.39	0.36	BART	Discriminative	Yes	-
T5-Medical (Siino, 2024)	0.63	0.3	0.5	0.48	T5-large-medical	Zero-Shot	No	-
USMBA-NLP (Fahfouh et al., 2024)	0.62	0.44	0.54	0.53	BERT base	Discriminative	Yes	-
SEME (Aguiar et al., 2024)	0.57	0.64	0.56	0.59	NLI-RoBERTa ensemble	Discriminative	Yes	-

Table 3: SemEval-2024 Task 2 Results, sorted by F1 (on the unperturbed subset of the test set), with Faithfulness (F), and Consistency (C)

models, with 17 out of the total, compared to 8 leveraging discriminative models. The F1 score suggests that GPT-4's performance is on par with considerably smaller models such as DeBERTa. However, a deeper evaluation using our novel metrics, especially Faithfulness, reveals a significant disparity, indicating that smaller models might be overfitting. This observation underscores the importance of employing these complementary metrics for a more comprehensive comparison of model capabilities. Despite the prevailing notion that larger models inherently perform better, this trend appears to be less pronounced than observed in this task's previous iteration (Jullien et al., 2023b), as illustrated in Figure 3. Notably, there seems to be a point of diminishing returns for model sizes between 7B and 70B, within the generative model category, shown in Figure 3. On average, models with sizes ranging from 7B to 70B parameters achieve +0.01 in F1 score but show decreases of -0.03 in Faithfulness and -0.01 in Consistency relative to models with more than 70B parameters. When compared to models with fewer than 7B parameters, these mid-sized models exhibit substantial improvements of +0.10 in F1 score, +0.40 in Faithfulness, and +0.19 in Consistency.

Table 4: Participant architectures by popularity, with average F1, Faithfulness (F) and Consistency (C)

Model	F1	F	C	Count
DeBERTa	0.76	0.76	0.75	5
Mistral 7B	0.75	0.84	0.69	4
Mixtral 8x7B	0.73	0.76	0.66	3
T5	0.66	0.36	0.53	3
Gemini Pro	0.70	0.77	0.68	3
GPT-4	**0.77**	**0.91**	0.76	2
SOLAR 10B	0.75	0.88	**0.77**	2
BERT base	0.62	0.44	0.54	1
Biolinkbert	0.72	0.59	0.64	1
BART	0.66	0.03	0.39	1
RoBERTa	0.57	0.64	0.56	1
Flan-T5 XL	0.76	0.84	0.75	1

Additionally, on average, generative models outperform discriminative ones across the board—with improvements observed in F1 scores (+0.025), Faithfulness (+0.15), and Consistency (+0.037), as depicted in Figure 2. Intriguingly, when comparing specific architectures, there is minimal correlation between model types and Faithfulness, Consistency, and F1, even though the top two performing systems in terms of F1 score are based on the Mixtral-8x7B-Instruct model (see Table 3).

5.2 Base F1 Performance

As previously mentioned the focus of this task extends beyond base performance. Nevertheless, it's noteworthy that the highest F1 score achieved in this iteration was 0.8 (Liu and Thoma, 2024; Guimarães et al., 2024) (FZI-WIM, Lisbon Computational Linguists) by two systems (Table 3). A figure that notably falls short of the previous iteration's top score of 0.856 (Zhou et al., 2023; Jullien et al., 2023b). This observed decline underscores a significant gap between the current capabilities of NLI systems and the performance required for practical application within clinical environments.

5.3 Faithfulness and Consistency

The overall average Faithfulness recorded at 0.719 significantly outperforms the average Consistency, which stands at 0.67. This disparity grows more pronounced within the subset of models within the top 10 F1 scores, where Average Faithfulness escalates to 0.835 and Average Consistency to 0.751.

Furthermore, a robust overall Spearman's correlation was identified between Consistency and F1 scores (0.8) and between Faithfulness and F1 scores (0.62). Intriguingly, this correlation inverts within the top 10 systems, where Spearman's Correlation between Consistency and F1 drops to -0.12, and between Faithfulness and F1 rises slightly to 0.319. Notably, the models with the highest Faithfulness (0.95) (Gema et al., 2024)(Edinburgh Clinical NLP) and Consistency (0.81) (Lee et al., 2024)(NYCU-NLP) scores achieve an average score of 0.84, surpassing systems ranked above them (with average scores of 0.81 and 0.78) yet both reporting a lower F1 score by -0.02. also Mandal and Modi (2024)(IITK) achieves a very high faithfulness of 0.9, while only managing an F1 of 0.69. These patterns underscore the limitation of F1 scores as sole indicators of model performance at the apex levels, accentuating the importance of considering Faithfulness and Consistency metrics in conjunction with F1.

The inversion of correlations among the top 10 models suggests a nuanced landscape of performance evaluation. While Consistency contributes broadly to high F1 scores, the top 10 models distinctly leverage Faithfulness, indicating that, at peak performance levels, perhaps accurate predictions rooted in correct premises are paramount over consistent responses to similar cases.

This phenomenon might also signify a ceiling effect for Consistency, suggesting that beyond a certain point, efforts to improve consistency do not translate into proportional performance gains. Such a scenario could inadvertently overshadow other critical model attributes like adaptability and nuanced comprehension, aspects more closely associated with Faithfulness. Alternatively, this situation could imply that models specifically optimized for F1 scores might inadvertently neglect Consistency, and to some degree, Faithfulness, as evidenced by the observed decline in their correlation with peak F1 scores.

Our analysis further elucidates the relationship between Consistency and Faithfulness in submitted systems, revealing an Overall Spearman Correlation of 0.708. This correlation slightly diminishes among the top 10 F1 scoring models to 0.39. While this represents a weaker correlation within the subset of the top 10 models, it importantly suggests the absence of a strict trade-off between Consistency and Faithfulness. Such a finding challenges the notion that improvements in one metric necessarily come at the expense of the other.

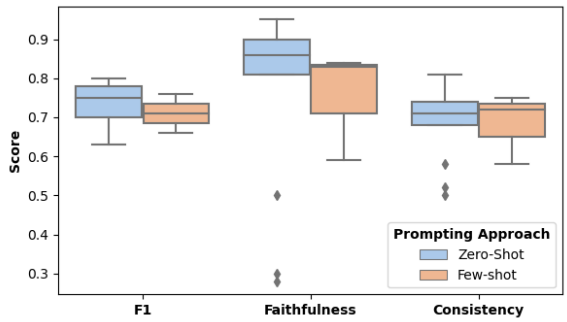


Figure 4: Comparative Analysis of F1, Consistency, and Faithfulness Across Prompting strategies

Among the participants, 4 out of 25 achieved a Faithfulness score of 0.9 or higher (Mandal and Modi, 2024; Liu and Thoma, 2024; Lee et al., 2024; Gema et al., 2024)(IITK, FZI-WIM, NYCU-NLP, Edinburgh Clinical NLP). Remarkably, only 1 out of 25 participants attained a Consistency score of 0.8 or higher (Lee et al., 2024)(NYCU-NLP). These results suggest a continued need for refining these models to achieve higher degrees of Faithfulness and Consistency if they are to be applied in real-world clinical environments.

5.4 Prompting Strategies

A variety of prompting strategies were used in the submitted systems. It is essential to acknowledge that variations in prompts can lead to significant differences in outcomes, even when employing the same architecture. For instance, within the Gemini Pro systems, a comparison between submissions by ALTINOK (2024)(D-NLP) and Kim et al. (2024)(Saama Technologies) from Saama Technologies reveals substantial disparities in performance metrics: F1 scores, Faithfulness, and Consistency differ by 0.09, 0.24, and 0.16, respectively. Similar patterns of variation were observed among submissions utilizing Mistral-based and T5-based approaches, underscoring the impact of prompting nuances.

Among the generative model submissions, 13 out of 16 employed a zero-shot approach, while the remaining three opted for few-shot prompting. Zero-shot prompting involves generating responses without any example-based guidance, relying solely on the model's pre-existing knowledge and the task description. Few-shot prompting, on the other hand, provides the model with one or more examples to guide its responses, traditionally anticipated to yield superior results.

Contrary to initial expectations, zero-shot prompting has shown a significant advantage, especially in achieving higher F1 scores and improving Faithfulness. Notably, four out of the top five models with the highest F1 scores utilized zero-shot techniques, as depicted in Figure 4. On average, zero-shot prompting yielded improvements of +0.025 in F1 score, +0.001 in Faithfulness, and +0.001 in Consistency, when compared to few-shot prompting methods.

Direct prompting is a straightforward method of querying a Language Model (LM). It involves posing a question to the model in a direct manner, without providing additional context or requesting intermediate steps. For example *"Given the CTR:* {Premise} *does the statement:* {Statement} *follow?"*

On the other hand, Chain of Thought (CoT) prompting represents a more elaborate technique designed to prompt the model to "show its work" by articulating the intermediate steps or reasoning that leads to its conclusion (Wei et al., 2022). This approach enables the model to break down the problem into smaller, more manageable parts, thereby facilitating more accurate or explainable predictions. For instance, the prompt could be structured as follows: *"Given the CTR:* {Premise} *and the statement:* {Statement}*, provide a step-by-step reasoning process to determine if the statement logically follows from the report."* Such a modification in the prompting strategy has been shown to produce significant differences in the model's outputs (Wei et al., 2022).

While direct prompting has been the predominant strategy among generative approaches, several teams have experimented with more nuanced strategies. Specifically, FZI-WIM (Liu and Thoma, 2024), IITK (Mandal and Modi, 2024), and Saama Technologies (Kim et al., 2024) have employed Chain of Thought prompting. Furthermore, IITK (Mandal and Modi, 2024) has also explored Tree of Thought (ToT) prompting. ToT prompting is an advanced technique aimed at improving the performance and interpretability of LMs, particularly in complex problem-solving tasks (Yao et al., 2023). It goes beyond the CoT approach by not merely listing reasoning steps linearly but by organizing these steps into a tree structure that represents different branches of reasoning or possible solutions. IITK (Mandal and Modi, 2024) applies this technique with the prompt *Imagine three different clinical experts are answering the question given below. All*

experts will write down first step of their thinking, then share it with the group. Then all experts will go on to the next step of their thinking. If any expert realises they're wrong at any point then they leave. They will continue till a definite conclusion is reached.. However, the ability to draw definitive conclusions about the relative efficacy of these prompting strategies is constrained given the considerable performance variability associated with each approach and the application of these strategies across diverse models, complicating efforts to ascertain the sources of performance gains or losses.

Two particularly intriguing prompting strategies emerged from the submissions. (Brutti-Mairesse, 2024)(CRCL) utilized an OPRO (Optimal Prompting for Response Optimization) technique (Yang et al., 2023), which leverages the model's ability to generate effective prompts from a small set of exemplars and prior instructions. This technique essentially tasks the model with creating its own instructions to tackle given problems. Additionally, (Lu and Kao, 2024) introduced a multi-agent debating framework, incorporating several custom agents with diverse expertise, including Biostatistics and Medical Linguistics, to enrich the model's output.

In summary, the submissions reveal a broad spectrum of prompting strategies, from zero-shot to more complex approaches like Tree of Thought and multi-agent frameworks. These strategies significantly influence model performance, underscoring the importance of prompt design in the development and evaluation of NLI systems. As the field progresses, further research is warranted to elucidate the optimal prompting strategies for enhancing model accuracy, reliability, and interpretability across various applications, in a controlled manner.

5.5 Fine-tuning strategies

Within the context of SemEval-2024 Task 2, a diverse array of fine-tuning strategies was employed across the 25 participating systems, revealing significant insights into their impact on model performance. Notably, 9 out of 25 systems, all of which were generative, did not undergo any form of fine-tuning. In contrast, 8 out of 25 systems were fine-tuned specifically on the NLI4CT-P training set, while the remaining 6 systems benefited from fine-tuning on additional datasets.

Interestingly, systems fine-tuned on the NLI4CT-P training set exhibited the lowest average performance across all three evaluated metrics, as detailed in Figure 5. Conversely, systems that underwent fine-tuning on external datasets demonstrated superior performance on all metrics, indicating a significant advantage of incorporating diverse training data.

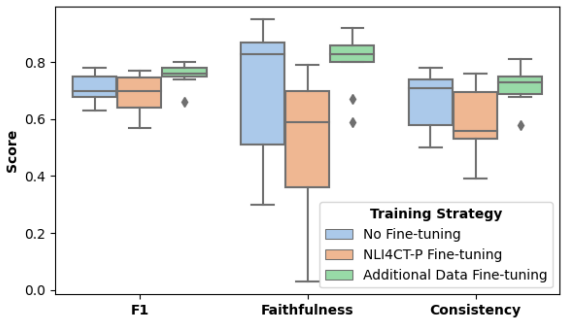


Figure 5: Comparative Analysis of F1, Consistency, and Faithfulness Across Training Strategies

The range of additional datasets leveraged for fine-tuning included various medical NLI datasets, such as MultiNLI, FeverNLI, ANLI, LingNLI, and WANLI, utilized by Zhang et al. (2024)(YNU-HPCC), and MedNLI by Verma and Raithel (2024)(DFKI-NLP). Moreover, some teams, including Sun et al. (2024)(LMU-BioNLP), Wang et al. (2024)(DKE-Research), Guimarães et al. (2024)(Lisbon Computational Lin- guists), Smilga and Alabiad (2024)(TüDuo), and Zhang et al. (2024)(YNU-HPCC), innovatively generated their data by applying interventions similar to those used in our task, thereby enriching their training material. Systems enhanced with additional data demonstrate significant improvements, achieving gains of +0.056 in F1 score, +0.132 in Faithfulness, and +0.062 in Consistency. These results suggest a substantial benefit from such tuning, particularly in terms of Faithfulness. This indicates that incorporating perturbed data into the training process not only enhances the model's inference ability but also significantly improves its reliability and adherence to the truthfulness of the clinical data it processes.

Instruction tuning emerged as a prevalent strategy, with datasets specifically crafted for this purpose by teams such as Liu and Thoma (2024)(FZI-WIM), Guimarães et al. (2024)(Lisbon Computational Lin- guists), Smilga and Alabiad (2024)(TüDuo), LUM-BIO, Wang et al. (2024)(DKE-Research), and (Lee et al., 2024)(NYCU-NLP). Notably, 3 out of the top 5 systems, as per F1 scores, employed instruction tuning, underscoring its effec-

tiveness in enhancing model performance, although notably producing minimal gains in consistency.

6 Related Work

The landscape of expert-annotated resources for clinical NLP is rich, with notable examples such as the TREC 2021 Clinical Track (Soboroff, 2021), which focuses on information retrieval from CTR data, highlighting eligibility criteria. Evidence Inference 2.0 (DeYoung et al., 2020) introduces a QA task alongside span selection based on CTR results, while the MEDNLI dataset (Romanov and Shivade, 2018) offers an entailment task using patient medical history notes. These datasets primarily aim to evaluate biomedical language understanding and reasoning. Despite neural architectures leading in biomedical NLI performance (Gu et al., 2021; DeYoung et al., 2020), challenges remain in quantitative reasoning and numerical operations within NLI (Ravichander et al., 2019; Galashov et al., 2019). Prior works experiment with biomedical pre-training strategies (Lee et al., 2020; Shin et al., 2020; Gu et al., 2021), and while ExaCT (Kiritchenko et al., 2010) automates information extraction from clinical trials, the integration of biomedical and numerical NLI effectively remains unaddressed. None of the aforementioned resources provide avenues for meaningful causal analysis, a gap NLI4CT-P aims to fill, through the application of targeted interventions and the introduction of novel evaluation metrics.

7 Conclusion

This study introduces the NLI4CT-P dataset and provides a comprehensive analysis of submissions to SemEval-2024 Task 2, underscoring the persistent challenges in Clinical Natural Language Inference (NLI) despite significant advancements in Large Language Models (LLMs). The incorporation of Faithfulness and Consistency metrics further highlights these challenges, shedding light on areas requiring additional focus, if these systems are to meet the requirements for real-world clinical implementation. Our key findings reveal that generative models markedly outperform discriminative models, particularly in terms of Faithfulness and Consistency. The utility of additional data is underscored, especially due to the limited size of the NLI4CT-P training set. Furthermore, our analysis reveals the substantial impact of prompting strategies on model performance, noting an intriguing preference for zero-shot approaches over few-shot methods. Additionally, mid-sized architectures, ranging between 7B and 70B parameters, demonstrate the potential to match or even exceed the performance of larger models (>70B) in F1 scores, Faithfulness, and Consistency, while being more resource and cost-effective. Conversely, models with fewer than 7B parameters face difficulties in achieving comparable results. We plan to perform a further analysis of the submitted systems' performance at an intervention level, identifying specific areas of weakness, such as numerical reasoning or handling longer premises, to refine and enhance Clinical NLI systems further.

8 Limitations

Despite not disclosing detailed specifics of the interventions, nor providing intervened training data, several participants generated their own interventions for data augmentation. As a result, some models were specifically trained on this intervened data. However, this approach raises concerns regarding their ability to generalize effectively to entirely new, unseen perturbations or adversarial datasets. The tailored training to specific interventions may limit the models' broader applicability and robustness on unseen perturbed or adversarial data.

9 Acknowledgments

This work was partially funded by the Swiss National Science Foundation (SNSF) project NeuMath (200021_204617), by the EPSRC grant EP/T026995/1 entitled "EnnCore: End-to-End Conceptual Guarding of Neural Architectures" under Security for all in an AI enabled society, by the CRUK National Biomarker Centre, and supported by the Manchester Experimental Cancer Medicine Centre.

References

Reem Abdel-Salam, Mary Adetutu Adewunmi, and Mercy Akinwale. 2024. Caresai at semeval-2024 task 2: Improving natural language inference in clinical trial data using model ensemble and data explanation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1916–1922, Mexico City, Mexico. Association for Computational Linguistics.

Mathilde Aguiar, Pierre Zweigenbaum, and Nona Naderi. 2024. Seme at semeval-2024 task 2: Comparing masked and generative language models on

natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 975–985, Mexico City, Mexico. Association for Computational Linguistics.

Abbas Akkasi, Adnan Khan, Mai A. Shaaban, Majid Komeili, and Mohammad Yaqub. 2024. iml at semeval-2024 task 2: Safe biomedical natural language inference for clinical trials with llm based ensemble inferencing. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 170–174, Mexico City, Mexico. Association for Computational Linguistics.

Duygu ALTINOK. 2024. D-nlp at semeval-2024 task 2: Evaluating clinical inference capabilities of large language models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 600–614, Mexico City, Mexico. Association for Computational Linguistics.

Nancy E Avis, Kevin W Smith, Carol L Link, Gabriel N Hortobagyi, and Edgardo Rivera. 2006. Factors associated with participation in breast cancer treatment clinical trials. *J Clin Oncol*, 24(12):1860–1867.

Hilda Bastian, Paul Glasziou, and Iain Chalmers. 2010. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9):e1000326.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Greg Brockman, Atty Eleti, Elie Georges, Joanne Jang, Logan Kilpatrick, Rachel Lim, Luke Miller, and Michelle Pokrass. 2023. ChatGPT and Whisper APIs. `https://openai.com/api/`. Accessed: April 3, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Clement Brutti-Mairesse. 2024. Crcl at semeval-2024 task 2: Simple prompt optimizations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 424–429, Mexico City, Mexico. Association for Computational Linguistics.

Abir Chakraborty. 2024. Rgat at semeval-2024 task 2: Biomedical natural language inference using graph attention network. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 116–122, Mexico City, Mexico. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Jiaxu Dao, Zhuoying Li, Xiuzhong Tang, Xiaoli Lan, and Junde Wang. 2024. Puer at semeval-2024 task 2: A biolinkbert approach to biomedical natural language inference. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 70–75, Mexico City, Mexico. Association for Computational Linguistics.

Spandan Das, Vinay Samuel, and Shahriar Noroozizadeh. 2024. Tldr at semeval-2024 task 2: T5-generated clinical-language summaries for deberta report analysis. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 507–516, Mexico City, Mexico. Association for Computational Linguistics.

Jay DeYoung, Eric P. Lehman, Benjamin E. Nye, Iain James Marshall, and Byron C. Wallace. 2020. Evidence inference 2.0: More data, better models. *ArXiv*, abs/2005.04177.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Anass Fahfouh, Abdessamad Benlahbib, Jamal Riffi, and Hamid Tairi. 2024. Usmba-nlp at semeval-2024 task 2: Safe biomedical natural language inference for clinical trials using bert. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 419–423, Mexico City, Mexico. Association for Computational Linguistics.

Alexandre Galashov, Jonathan Schwarz, Hyunjik Kim, Marta Garnelo, David Saxton, Pushmeet Kohli, S. M. Ali Eslami, and Yee Whye Teh. 2019. Meta-learning surrogate models for sequential decision making. *CoRR*, abs/1903.11907.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Aryo Gema, Giwon Hong, Pasquale Minervini, Luke Daines, and Beatrice Alex. 2024. Edinburgh clinical nlp at semeval-2024 task 2: Fine-tune your model unless you have access to gpt-4. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1905–1915, Mexico City, Mexico. Association for Computational Linguistics.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Lisa Grossman Liu, Raymond H Grossman, Elliot G Mitchell, Chunhua Weng, Karthik Natarajan, George Hripcsak, and David K Vawdrey. 2021. A deep database of medical abbreviations and acronyms for natural language processing. *Scientific Data*, 8(1):1–9.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Artur Guimarães, Bruno Martins, and João Magalhães. 2024. Lisbon computational linguists at semeval-2024 task 2: Using a mistral-7b model and data augmentation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1270–1277, Mexico City, Mexico. Association for Computational Linguistics.

Antonio J Jimeno-Yepes, Bridget T McInnes, and Alan R Aronson. 2011. Exploiting mesh indexing in medline to generate a data set for word sense disambiguation. *BMC bioinformatics*, 12(1):1–14.

Qiao Jin, Jinling Liu, and Xinghua Lu. 2019. Deep contextualized biomedical abbreviation expansion. *arXiv preprint arXiv:1906.03360*.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023a. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023b. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2216–2226, Toronto, Canada. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama ai research at semeval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation*.

Hwanmun Kim, Kamal raj Kanakarajan, and Malaikannan Sankarasubbu. 2024. Saama technologies at semeval-2024 task 2: Three-module system for nli4ct enhanced by llm-generated intermediate labels. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1423–1445, Mexico City, Mexico. Association for Computational Linguistics.

Svetlana Kiritchenko, Berry De Bruijn, Simona Carini, Joel Martin, and Ida Sim. 2010. Exact: automatic extraction of clinical trial characteristics from journal publications. *BMC medical informatics and decision making*, 10(1):1–17.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Lung-Hao Lee, Chen-Ya Chiou, and Tzu-Mi Lin. 2024. Nycu-nlp at semeval-2024 task 2: Aggregating large language models in biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1465–1472, Mexico City, Mexico. Association for Computational Linguistics.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Hanmeng Liu, Leyang Cui, Jian Liu, and Yue Zhang. 2021. Natural language inference in context-investigating contextual reasoning over long texts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13388–13396.

Jin Liu and Steffen Thoma. 2024. Fzi-wim at semeval-2024 task 2: Self-consistent cot for complex nli in biomedical domain. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1259–1269, Mexico City, Mexico. Association for Computational Linguistics.

Yu-An Lu and Hung-Yu Kao. 2024. 0x.yuan at semeval-2024 task 2: Agents debating can reach consensus and produce better outcomes in medical nli task. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 305–310, Mexico City, Mexico. Association for Computational Linguistics.

Shreyasi Mandal and Ashutosh Modi. 2024. Iitk at semeval-2024 task 2: Exploring the capabilities of llms for safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1386–1393, Mexico City, Mexico. Association for Computational Linguistics.

Jennifer Marks, MohammadReza Davari, and Leila Kosseim. 2024. Clac at semeval-2024 task 2: Faithful clinical trial inference. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1683–1687, Mexico City, Mexico. Association for Computational Linguistics.

Jordan Meadows, Marco Valentino, Damien Teney, and Andre Freitas. 2023. A symbolic framework for systematic evaluation of mathematical reasoning with transformers. *arXiv preprint arXiv:2305.12563*.

Marius Micluța-Câmpeanu, Claudiu Creanga, Ana-Maria Bucur, Ana Sabina Uban, and Liviu P. Dinu. 2024. Unibuc at semeval-2024 task 2: Tailored prompting with solar for clinical nli. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 573–582, Mexico City, Mexico. Association for Computational Linguistics.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. The effect of natural distribution shift on question answering models. In *International Conference on Machine Learning*, pages 6905–6916. PMLR.

Shantanu Nath and Ahnaf Mozib Samin. 2024. Bd-nlp at semeval-2024 task 2: Investigating generative and discriminative models for clinical inference with knowledge augmentation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1291–1297, Mexico City, Mexico. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? *CoRR*, abs/2103.07191.

Kayur Patel, James Fogarty, James A Landay, and Beverly Harrison. 2008. Investigating statistical machine learning as a tool for software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 667–676.

Ahmad Pesaranghader, Stan Matwin, Marina Sokolova, and Ali Pesaranghader. 2019. deepbiowsd: effective deep neural word sense disambiguation of biomedical text data. *Journal of the American Medical Informatics Association*, 26(5):438–446.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Penstein Rosé, and Eduard H. Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. *CoRR*, abs/1901.03735.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv:1808.06752*.

Julia Rozanova, Marco Valentino, and Andre Freitas. 2023. Estimating the causal effects of natural logic features in neural nli models. *arXiv preprint arXiv:2305.08572*.

Hoo-Chang Shin, Yang Zhang, Evelina Bakhturina, Raul Puri, Mostofa Patwary, Mohammad Shoeybi, and Raghav Mani. 2020. Biomegatron: Larger biomedical domain language model. *arXiv preprint arXiv:2010.06060*.

Marco Siino. 2024. T5-medical at semeval-2024 task 2: Using t5 medical embedding for natural language inference on clinical trial data. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 40–46, Mexico City, Mexico. Association for Computational Linguistics.

Veronika Smilga and Hazem Alabiad. 2024. Tüduo at semeval-2024 task 2: Flan-t5 and data augmentation for biomedical nli. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 723–730, Mexico City, Mexico. Association for Computational Linguistics.

Ian Soboroff. 2021. Overview of trec 2021. In *30th Text REtrieval Conference. Gaithersburg, Maryland*.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schölkopf, and Mrinmaya Sachan. 2022. A causal framework to quantify the robustness of mathematical reasoning with language models. *arXiv preprint arXiv:2210.12023*.

Zihang Sun, Danqi Yan, Anyi Wang, Tanalp Agustoslu, Qi Feng, Chengzhi Hu, Longfei Zuo, Shijia Zhou, Hermine Kleiner, Pingjun Hong, Suteera Seeha, Sebastian Loftus, Anna Barwig, Oliver Kraus, Jona Volohonsky, Yang Sun, Leopold Martin, Lena Altinger, Jing Wang, and Leon Weber. 2024. Lmu-bionlp at semeval-2024 task 2: Large diverse ensembles for robust clinical nli. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1587–1593, Mexico City, Mexico. Association for Computational Linguistics.

Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. 2020. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1):1–10.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. *arXiv preprint arXiv:1804.08117*.

Bhuvanesh Verma and Lisa Raithel. 2024. Dfki-nlp at semeval-2024 task 2: Towards robust llms using data perturbations and minmax training. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 668–682, Mexico City, Mexico. Association for Computational Linguistics.

Juraj Vladika and Florian Matthes. 2023. Sebis at semeval-2023 task 7: A joint system for natural language inference and evidence retrieval from clinical trial reports. In *Proceedings of the 17th International Workshop on Semantic Evaluation*.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2021. Measure and improve robustness in nlp models: A survey. *arXiv preprint arXiv:2112.08313*.

Yuqi Wang, Zeqiang Wang, Wei Wang, Qi Chen, Kaizhu Huang, Anh Nguyen, and Suparna De. 2024. Dke-research at semeval-2024 task 2: Incorporating data augmentation with generative models and biomedical knowledge to enhance inference robustness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 88–94, Mexico City, Mexico. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Xiaoyu Xing, Zhijing Jin, Di Jin, Bingning Wang, Qi Zhang, and Xuanjing Huang. 2020. Tasty burgers, soggy fries: Probing aspect robustness in aspect-based sentiment analysis. *arXiv preprint arXiv:2009.07964*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Sicheng YU, Jing JIANG, Hao ZHANG, Yulei NIU, Qianru SUN, and Lidong BING. 2022. Interventional training for out-of-distribution natural language understanding.

Rengui Zhang, Jin Wang, and Xuejie Zhang. 2024. Ynuhpcc at semeval-2024 task 2: Applying deberta-v3-large to safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 772–778, Mexico City, Mexico. Association for Computational Linguistics.

Yuxuan Zhou, Ziyu Jin, Meiwei Li, Miao Li, Xien Liu, Xinxin You, and Ji Wu. 2023. Thifly research at semeval-2023 task 7: A multi-granularity system for ctr-based textual entailment and evidence retrieval. In *Proceedings of the 17th International Workshop on Semantic Evaluation*.

A Intervention Prompts

A.1 Contradictory Rephrasing prompt

Your task is to provide 3 contradictory statements, given an original statement.

(Instructions) Ensure that the contradictory statements are factually opposed to the original statement. Do not mention the original statement in the contradictory statements. Use formal and straightforward language when writing the new statements, and avoid unusual or overly descriptive language. Make sure to retain the names 'Primary Clinical Trial' and 'Secondary Clinical Trial' in the contradictory statements, these names must be present in every statement. Provide 3 different options in a consistent JSON format with keys 'Statement_1', 'Statement_2', and 'Statement_3' followed by their respective paraphrased statements.

(Examples) 1. [original statement]:"the secondary trial requires patients to be over a certain age, but the primary trial does not specify an age range for participation." [ideal output]: "the secondary trial does not give an age limit for patients to participate, but patients must be between the age of 12-34 to be eligible for the primary trial"

2. [original statement]:"a patient that has received an organ transplant within the last month, and is still bedridden would be excluded from the primary trial but may be eligible for the secondary trial" [ideal output]: "a patient that has received an liver transplant in the last week, with an ECOG score of 4 would be eligible for the primary trial but excluded from the secondary trial"

3.[original statement]: "Women with Newly diagnosed stage IV breast cancer, confirmed as ER+ Considering a mastectomy are eligible for the primary trial" [ideal output]: "Women recently diagnosed with stage 4 ER-positive breast cancer and contemplating a mastectomy are excluded from the Primary Clinical Trial"

Input:

A.2 Paraphrasing prompt

Your task is to provide 3 paraphrased statements, given an original statement.

(Instructions) Use formal and straighforward language when writing the new statements, and avoiding unusual or overly descriptive language. Make sure to retain the name 'Primary Clinical Trial' in the statements, this name must be present in every statement. Provide 3 different options in a consistent JSON format with keys 'Statement_1', 'Statement_2', and 'Statement_3' followed by their respective paraphrased statements.

(Examples) 1. [original statement]:"the primary trial does not specify an age range for participation." [ideal output]: "patients aged between 30-60 years old can be eligible for the primary trial"

2. [original statement]:"a patient that has received an organ transplant within the last month, and is still bedridden would be excluded from the primary trial" [ideal output]: "a patient that has received an liver transplant in the last week, with an ECOG score of 4 would be excluded from the primary trial"

3.[original statement]: "Women with Newly diagnosed stage IV breast cancer, confirmed as ER+ Considering a mastectomy are eligible for the primary trial" [ideal output]: "Women recently diagnosed with stage 4 ER-positive breast cancer and contemplating a mastectomy are suitable for the Primary Clinical Trial"

Input:

A.3 Numerical Paraphrasing prompt

Your task is to modify the numerical values and units in an original statement while maintaining its original meaning, to generate 3 new statements.

(Instructions) Do not paraphrase the statements, You can only change numerical values or units, if you change the units you must also convert the measurement values. Provide 3 different options in a consistent JSON format with keys 'Statement_1', 'Statement_2', and 'Statement_3' followed by their respective paraphrased statements.

(Examples) 1. [original statement]: "Over 6 weeks of TAK-228 Plus Tamoxifen treatment patients in the primary trial experienced a 5% reduction in the Percentage of cells with Ki67 expression" [ideal output]: "Over 42 days of TAK-228 Plus Tamoxifen treatment patients in the primary trial experienced a 5% reduction in the Percentage of cells with Ki67 expression"

2.[original statement]: "in the primary trial there were 10 times the number of Hepatotoxicity cases as there were cases of hypertension and Pancreatectomy" [ideal output]: "in the primary trial there were 1000% the number of Hepatotoxicity cases as there were cases of hypertension and Pancreatectomy"

3.[original statement]: "2/73 the primary trial participants, and 0/1674 the secondary trial participants suffered an Acute myocardial infarction " [ideal output]: "2.74% the primary trial participants, and 0% the secondary trial participants suffered an Acute myocardial infarction "

Input:

A.4 Numerical Contradictory Rephrasing prompt

Your task is to modify the numerical values and units in an original statement to contradict the original statement, to generate 3 new statements.

(Instructions) Do not paraphrase the statements, You can only change numerical values or units, if you change the units you must also convert the measurement values. Provide 3 different options in a consistent JSON format with keys 'Statement_1', 'Statement_2', and 'Statement_3' followed by their respective paraphrased statements.

(Examples) 1. [original statement]: "Over 6 weeks of TAK-228 Plus Tamoxifen treatment patients in the primary trial experienced a 5% reduction in the Percentage of cells with Ki67 expression"

[ideal output]: "Over 50 days of TAK-228 Plus Tamoxifen treatment patients in the primary trial experienced a 105% reduction in the Percentage of cells with Ki67 expression"

2.[original statement]: "in the primary trial there were 10 times the number of Hepatotoxicity cases as there were cases of hypertension and Pancreatectomy" [ideal output]: "in the primary trial there were 30% the number of Hepatotoxicity cases as there were cases of hypertension and Pancreatectomy"

3.[original statement]: "2/73 the primary trial participants, and 0/1674 the secondary trial participants suffered an Acute myocardial infarction " [ideal output]: "9.74% the primary trial participants, and 8% the secondary trial participants suffered an Acute myocardial infarction "

Input:

SemEval-2024 Task 1: Semantic Textual Relatedness for African and Asian Languages

Nedjma Ousidhoum[1]*, Shamsuddeen Hassan Muhammad[2]* , Mohamed Abdalla, Idris Abdulmumin[3], Ibrahim Said Ahmad[4], Sanchit Ahuja[5], Alham Fikri Aji[6], Vladimir Araujo[7], Meriem Beloucif[8], Christine De Kock[9], Oumaima Hourrane, Manish Shrivastava[10], Thamar Solorio[6], Nirmal Surange[10], Krishnapriya Vishnubhotla[11], Seid Muhie Yimam[12], Saif M. Mohammad[13]

[1]Cardiff University, [2]Imperial College London, [3]Data Science for Social Impact Research Group, University of Pretoria, [4]Institute For Experiential AI, Northeastern University, [5]BITS Pilani, [6]MBZUAI, [7]KU Leuven, [8]Uppsala University, [9]The University of Merlbourne, [10]IIIT Hyderabad, [11]University of Toronto, [12]Universität Hamburg, [13]National Research Council Canada.

Contact: OusidhoumN@cardiff.ac.uk, s.muhammad@imperial.ac.uk

Abstract

We present the first shared task on Semantic Textual Relatedness (STR). While earlier shared tasks primarily focused on semantic similarity, we instead investigate the broader phenomenon of semantic relatedness across 14 languages: *Afrikaans, Algerian Arabic, Amharic, English, Hausa, Hindi, Indonesian, Kinyarwanda, Marathi, Moroccan Arabic, Modern Standard Arabic, Punjabi, Spanish,* and *Telugu*. These languages originate from five distinct language families and are predominantly spoken in Africa and Asia – regions characterised by the relatively limited availability of NLP resources. Each instance in the datasets is a sentence pair associated with a score that represents the degree of semantic textual relatedness between the two sentences. Participating systems were asked to rank sentence pairs by their closeness in meaning (i.e., their degree of semantic relatedness) in the 14 languages in three main tracks: (a) supervised, (b) unsupervised, and (c) crosslingual. The task attracted 163 participants. We received 70 submissions in total (across all tasks) from 51 different teams, and 38 system description papers. We report on the best-performing systems as well as the most common and the most effective approaches for the three different tracks.

1 Introduction

Defining the relationship between two units of text is an important component of constructing text representations. Within this context, semantic textual relatedness (STR) aims to capture the degree to which two linguistic units (e.g., words or sentences, etc.) are close in meaning (Mohammad and Hirst, 2012). Two units may be related in a variety of different ways (e.g., by expressing the same view, originating from the same time period, elaborating on each other, etc.). On the other hand, semantic textual similarity (STS) considers only a narrow view of the relationship that may exist between texts (such as equivalence or paraphrase) which does not incorporate other dimensions of relatedness such as entailment, topic or view similarity, or temporal relations (Abdalla et al., 2023). For example, *'I am feeling sick.'* and *'Get well soon!'* would receive a low similarity score, despite the two being very related. In this shared task, we investigate the broader concept of semantic textual relatedness. STR is central to understanding meaning in text (Hasan and Halliday, 1976; Miller and Charles, 1991; Morris and Hirst, 1991) and its automation can benefit various downstream tasks such as evaluating sentence representation methods, question answering, and summarisation (Abdalla et al., 2023; Wang et al., 2022).

Prior shared tasks (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017) have mainly focused on textual similarity. In this work, we provide participants with SemRel (Ousidhoum et al., 2024), a collection of 14 newly curated monolingual STR datasets for Afrikaans (`afr`), Amharic (`amh`), Modern Standard Arabic (`arb`), Algerian Arabic (`arq`), Moroccan Arabic (`ary`), English (`eng`), Spanish (`esp`), Hausa (`hau`), Hindi (`hin`), Indonesian (`ind`), Kinyarwanda (`kin`), Marathi (`mar`), Punjabi (`pun`) and Telugu (`tel`). The datasets are composed of sentence pairs, each assigned a relatedness score between 0 (completely

*Equal contribution from first and second authors, authors 3 to 16 are alphabetically ordered.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1963–1978

Lang.	Family	Train	Dev	Test
afr	Indo-European	-	375	375
amh	Afro-Asiatic	992	95	171
arb	Afro-Asiatic	-	32	595
arq	Afro-Asiatic	1,261	97	583
ary	Afro-Asiatic	925	70	427
eng	Indo-European	5,500	250	2,600
esp	Indo-European	1,562	140	600
hau	Afro-Asiatic	1,763	212	603
hin	Indo-European	-	288	968
ind	Austronesian	-	144	360
kin	Niger-Congo	778	102	222
mar	Indo-European	1,200	293	298
pan	Indo-European	-	638	242
tel	Dravidian	1,170	130	297

Table 1: The language families and data split sizes of the different datasets. Datasets with no traning sets were only used in tracks B and C.

unrelated) and 1 (maximally related) with a large range of expected relatedness values. The pairs of sentences were first selected from pre-existing datasets covering various topics and formality levels, e.g., news data, Wikipedia, and conversational data. To generate the relatedness scores, the sentence pairs were then annotated by native speakers who performed comparisons between different pairs of sentences using Best–Worst Scaling (BWS) (Louviere and Woodworth, 1991; Kiritchenko and Mohammad, 2017a). The shared task included three main tracks: (1) supervised, (2) unsupervised, and (3) cross-lingual.

Each team could provide submissions for one, two, or all of the tracks in one or more languages. Our official evaluation metric was the Spearman rank correlation coefficient, which captures how well the system-predicted rankings of test instances aligned with human judgments. Our task attracted 163 participants, received 70 final submissions from 51 different teams, and 38 teams submitted system description papers. Track A (supervised) received the largest number of submissions: 40, followed by 18 submissions for track B (unsupervised) and 12 for track C (crosslingual). Most teams participated in multiple languages (more than eight on average). All of the task details and resources are available on the task website.[1]

2 Related Work

The field of semantic textual relatedness in natural language processing covers a variety of approaches and techniques designed to measure the closeness in meaning between units of text, specifically words (Miller, 1994) or sentences (Abdalla et al., 2023).

Most prior shared tasks focus on semantic textual similarity, a narrower subset of relatedness, and often only cover high-resource languages such as English (Agirre et al., 2012, 2013, 2014, 2015, 2016), Arabic, German, Spanish, and Turkish (Cer et al., 2017) with few exceptions such as Armendariz et al. (2020) who also included Slovene, Finnish, and Croatian.

By comparison, this shared task focuses on sentence-level STR in various low-resource languages. To our knowledge, the only corpora specially designed for semantic textual relatedness between pairs of sentences was created by Abdalla et al. (2023) for English. The core of Abdalla et al. (2023) approach served as the model for data annotations added to new ways of data collection–curation for several less-resourced languages.

3 Data

3.1 Data Collection

A key step in the data creation process was identifying text sources for each language and selecting sentence pairs. This was particularly challenging for low-resource languages such as Hausa, Telugu, or Algerian Arabic. Since most SemRel languages are low-resource, the domain, (in)formality, and diversity of the sentence pairs were highly dependent on the publicly available corpora. We aimed to collect datasets with average-length sentences, free of offensive utterances, and as diverse as possible. Thus, data instances were extracted for each language using a tailored combination of heuristics such as lexical overlap and paraphrases. We used further pre-processing, post-processing, and data analysis methods to avoid incoherence and unnaturalness.

Since arbitrarily selecting sentences and pairing them would lead to many unrelated instances, we relied on the following heuristics to pair sentences and ensure that the pairs would exhibit relatedness scores varying from completely unrelated to very related:

1. **Lexical Overlap** Select sentences with various proportions of lexical overlap, i.e., one or more words/tokens in common, with or without using TF/IDF normalisation.
2. **Contiguity/Entailment** Select adjacent pairs of sentences in a paragraph or a social media

[1] https://semantic-textual-relatedness.github.io

Language	afr	amh	arb	arq	ary	eng	esp	hau	hin	ind	kin	mar	pun	tel
#Annotators	2	4	2-3	2	2	2-4	2-4	2-4	4	2	2	2-3	2	4
SHR train/dev	0.85	0.89	0.86	0.64	0.77	0.80	0.70	0.74	0.93	0.68	0.74	0.92	0.65	0.79
SHR test	0.85	0.89	0.86	0.64	0.77	0.80	0.70	0.74	0.93	0.68	0.74	0.92	0.65	0.79

Table 2: SHR (split-half reliability) scores for each of the created dataset splits and numbers of annotators per tuple (#Annotators).

thread, i.e., sentences that appear one after the other.

3. **Paraphrases or Machine Translation (MT) Paraphrases** Select pairs of sentences from paraphrase or MT data. For MT, we pivot across the translation and back to the source language to generate a new sentence and pair it with the original.
4. **Random selection** Random pairs of sentences are selected.

We elaborate on the detailed data collection and processing steps in Ousidhoum et al. (2024).

3.2 Data Annotation

As the notions of *related* and *unrelated* do not have clear boundaries with no unanimous definition in the literature, we use comparative annotations and rely on the intuitions of fluent speakers for each language to choose between sentence pairs. Therefore, instead of relying on vague class definitions, we capture common perceptions of semantic relatedness (i.e., what is believed by the vast majority) rather than "correct" or "right" rankings.

We used Best–Worst Scaling (BWS) (Louviere and Woodworth, 1991; Kiritchenko and Mohammad, 2017a), a form of comparative annotation that avoids various biases of traditional rating scales, to annotate our data instances and generate an ordinal ranking of instances. In BWS, annotators are given n items (an n-tuple, where $n > 1$ and commonly $n = 4$). They are asked which item is the *best* (highest in terms of the property of interest) and which is the *worst* (lowest in terms of the property of interest). When working on 4-tuples, best–worst annotations are particularly efficient because each best and worst annotation will reveal the order of five of the six-item pairs. Real-valued scores of association between the items and the property of interest can be determined using simple arithmetic on the number of times an item was chosen best and the number of times it was chosen worst (Orme, 2009; Flynn and Marley, 2014). It has been empirically shown that annotations for $2N$ 4-tuples are sufficient for obtaining reliable scores (where N is the number of items) (Louviere and Woodworth, 1991; Kiritchenko and Mohammad, 2016). Kiritchenko and Mohammad (2017b) showed through empirical experiments that BWS produces more reliable and discriminating scores than those obtained using rating scales. (See (Kiritchenko and Mohammad, 2016, 2017b) for further details on BWS.) We generated tuples using the BWS scripts provided by Kiritchenko and Mohammad (2017a)[2].

We report the number of annotators and the split-half reliability (SHR) scores (Cronbach, 1951; Kuder and Richardson, 1937) for each of the datasets in Table 2. SHR measures the degree to which repeating the annotations results in similar relative rankings of the instancesOverall the scores in Table 2 vary between 0.64 and 0.96, which indicates a high annotation reliability.

4 Task Description

In this task, we aim to predict the semantic textual relatedness (STR) of sentence pairs. Participants had to rank sentence pairs by their degree of semantic relatedness which varies between 0 (unrelated) and 1 (closely related). Each team could provide submissions for one, two, or all of the tracks presented below.

4.1 Track A: Supervised

Participants were to submit systems trained on the labeled training datasets provided. Participating teams were allowed to use any publicly available datasets (e.g., other relatedness and similarity datasets or datasets in any other languages). However, they had to report on additional data they used, and ideally report how each resource impacted the final results.

4.2 Track B: Unsupervised

Participants were to submit systems that were developed without the use of any labeled datasets

[2]https://saifmohammad.com/WebPages/BestWorst.html

#	Track A (Supervised)			Track B (Unsupervised)			Track C (Crosslingual)	
	Team	Score		Team	Score		Team	Score
			*	Lexical Overlap	0.456			
*	baseline (LaBSE)	0.762	*	baseline (XLMR)	0.353	*	baseline (LaBSE)	0.579
1	AAdam	0.800		SATLab	0.543		AAdaM	0.650
2	NRK	0.781		MasonTigers	0.514		UAlberta	0.589
3	PEAR	0.758		HW–TSC	0.482		silp_nlp	0.566
4	silp_nlp	0.740		UAlberta	0.481		MaiNLP	0.499
5	NLP_1@SSN	0.740		silp_nlp	0.400		ustcctsu	0.445

Table 3: Top 5 submissions per track. See Appendix for paper information about the different teams. * shows baseline results using lexical overlap, XLMR and LaBSE reported in the SemRel dataset paper (Ousidhoum et al., 2024).

pertaining to semantic relatedness or semantic similarity between units of text more than two words long in any language. The use of unigram or bigram relatedness datasets (from any language) was permitted.

4.3 Track C: Cross-lingual

Participants were to submit systems that were developed without the use of any labeled semantic similarity or semantic relatedness datasets in the target language and with the use of labeled dataset(s) from at least one other language. Using labeled data from another track was mandatory for a submission to this track.

4.4 Official Evaluation Metric

The official evaluation metric for this task is the Spearman rank correlation coefficient, which captures how well the system-predicted rankings of test instances align with human judgments. We provided the participants with an evaluation script on GitHub page[3].

4.5 Task Organisation

We released some pilot datasets before the start of the shared task for participants to have a better understanding of the task (i.e., the datasets, the languages involved, and the labels) and provided the participants with a starter kit on GitHub.

5 Evaluation

5.1 Our baselines

In Table 3, we report a simple lexical overlap baseline which consists of the Dice coefficient between two sentences A and B: the number of unique unigrams occurring in both sentences, adjusted by their lengths (Abdalla et al., 2023):

$$\frac{2 \times |unigram(A) \cap unigram(B)|}{|unigram(A) + unigram(B)|} \quad (1)$$

In addition, we used LaBSE (Label Agnostic BERT Sentence Embeddings) (Feng et al., 2020) which can map 109 languages into a shared vector space. With the embeddings covering all the SemRel languages, we report baseline results using the default hyperparameters set in the sentence-transformers repository[4]. We used:

- the predefined setup without further fine-tuning,
- the LaBSE model further fine-tuned on our training data using a cosine similarity loss.

For the crosslingual baselines, we fine-tuned LaBSE on the English training set and tested on all the other datasets except English while using the Spanish training set to fine-tune LaBSE when testing on English. We elaborate on the detailed baseline experiment in (Ousidhoum et al., 2024)

5.2 Participating Systems and Results

5.3 Participant Overview

During the evaluation phase, 163 people registered for the competition. Of these, 51 teams made 70 final submissions across tracks [5]. Track A received 40 final submissions, track B received 12 submissions, and track C received 18. For track A, most participants submitted systems for at least eight languages. We report the top–5 performing systems in all tracks in Table 3.

[3] https://github.com/semantic-textual-relatedness/Semantic_Relatedness_SemEval2024

[4] https://github.com/UKPLab/sentence-transformers

[5] The details can be found in the Appendix.

Rank	Team	amh	arq	ary	eng	esp	hau	kin	mar	tel	Average
1	AAdaM (Zhang et al., 2024)	0.867	0.662	0.835	0.848	**0.740**	0.724	0.779	0.894	0.848	0.800
2	NRK (Nguyen and Thin, 2024)	0.864	0.674	0.827	0.833	0.690	0.672	0.757	0.879	0.834	0.781
*	SemRel baseline (LaBSE)	0.789	0.847	0.761	0.830	0.702	0.693	0.725	0.881	0.817	0.762
3	PEAR (Jørgensen, 2024)	0.834	0.463	0.815	0.848	0.710	0.694	0.772	0.856	0.827	0.758
4	silp_nlp (Singh et al., 2024)	0.837	0.594	0.808	0.845	0.658	0.724	0.485	0.863	0.843	0.740
5	NLP_1@SSN (B et al., 2024)	-	0.623	0.745	0.835	0.705	0.628	0.723	0.871	0.789	0.740
6	UAlberta (Shi et al., 2024)	0.854	0.464	0.497	0.853	0.705	0.735	0.641	0.890	0.857	0.722
7	MBZUAI-UNAM (Ortiz-Barajas et al., 2024)	0.840	0.541	0.786	0.832	0.697	0.670	0.458	0.867	0.785	0.720
8	INGEOTEC (Moctezuma et al., 2024)	0.702	0.566	0.811	0.809	0.678	0.576	0.630	0.784	0.801	0.706
9	HausaNLP (Salahudeen et al., 2024)	0.353	0.587	0.834	0.794	0.723	0.594	0.633	0.837	0.800	0.684
10	KINLP	-	0.471	0.779	0.740	0.581	0.616	0.763	0.749	0.754	0.682
11	BITS Pilani (Venkatesh and Raman, 2024)	0.800	0.510	0.444	0.832	0.656	0.508	0.518	0.842	0.814	0.658
12	OZemi (Takahashi et al., 2024)	0.781	0.371	0.445	0.805	0.620	0.620	0.567	0.862	0.782	0.650
13	Text Mining (Keinan, 2024)	0.713	0.443	0.701	0.720	0.661	0.543	0.413	0.778	0.706	0.631
14	MasonTigers (Goswami et al., 2024)	0.785	0.400	0.376	0.836	0.651	0.477	0.367	0.818	0.802	0.612
15	YSP (Aali et al., 2024)	0.643	0.402	-	0.819	0.635	0.387	0.315	0.689	0.643	0.567
16	IITK (Basak et al., 2024)	0.550	0.339	0.358	0.808	0.591	0.219	0.138	0.666	0.282	0.439
17	YNUNLP2023 (Li et al., 2024b)	0.789	0.235	0.092	0.557	0.404	0.269	0.186	0.544	0.617	0.410
NR	PALI	**0.889**	0.679	**0.863**	**0.860**	0.724	**0.764**	0.813	**0.911**	0.864	**0.819**
NR	king001	0.888	**0.682**	0.860	0.843	0.721	0.747	**0.817**	0.897	0.853	0.812
NR	saturn	0.845	0.578	0.798	-	-	0.699	0.755	0.873	**0.873**	0.774
NR	UMBCLU (Roy Dipta and Vallurupalli, 2024)	-	-	0.745	0.838	0.721	0.640	0.681	0.841	0.682	0.733
NR	SemanticCUETSync (Hossain et al., 2024)	-	-	-	0.822	0.677	-	-	0.870	0.820	0.796
NR	NLP-LISAC (Benlahbib et al., 2024)	-	0.604	0.789	0.835	0.717	-	-	-	-	0.736
NR	Unknown	-	-	-	0.831	-	-	-	0.882	0.841	0.852
NR	BpHigh	-	-	-	0.809	-	-	-	0.875	0.769	0.819
NR	Sharif_STR (Ebrahimi et al., 2024)	-	0.380	-	0.827	0.673	-	-	-	-	0.441
NR	CAILMD-23 (Sonavane et al., 2024)	-	-	-	0.823	-	-	-	0.871	-	0.847
NR	WarwickNLP (Ebrahim and Joy, 2024)	-	-	0.816	0.842	-	-	-	-	-	0.829
NR	GIL-IIMAS UNAM	-	-	-	0.830	0.731	-	-	-	-	0.780
NR	msiino	-	-	-	0.809	0.611	-	-	-	-	0.710
NR	NLU-STR (Malaysha et al., 2024)	-	0.525	0.832	-	-	-	-	-	-	0.678
NR	Tübingen-CL (Zhang and Çöltekin, 2024)	-	-	-	0.850	-	-	-	-	-	0.850
NR	Pinealai (Eponon and Ramos Perez, 2024)	-	-	-	0.837	-	-	-	-	-	0.837
NR	gds142	-	-	-	-	-	-	-	-	0.826	0.826
NR	LuisRamos07	-	-	-	0.822	-	-	-	-	-	0.822
NR	VerbaNexAI Lab (Morillo et al., 2024)	-	-	-	0.819	-	-	-	-	-	0.819
NR	Fired_from_NLP (Shanto et al., 2024)	-	-	-	0.810	-	-	-	-	-	0.810
NR	Roronoa_Zoro	-	-	-	0.810	-	-	-	-	-	0.810
NR	NLP_STR_teamS (Su and Zhou, 2024)	-	-	-	0.809	-	-	-	-	-	0.809
NR	DataJo	-	0.356	-	-	-	-	-	-	-	0.356

Table 4: Track A results. The best results are in bold, and NR stands for *not ranked*. As the methods are highly language-dependent, we only rank teams that participated in at least 8 sub-tracks, but we highlight in blue the best results achieved by non-ranked teams. (Non-ranked teams are sorted based on the number of languages they participated in.)

5.4 Task A: Supervised

5.4.1 Best Performing Systems

AAdaM They opted for data augmentation by translating the English SemRel dataset and STSB (semantic similarity) to create and augment data in other languages. The team explored both fine-tuning and adapter-based tuning. Given a target language, they first fine-tuned the cross-encoder-based AfroXLMR model (Alabi et al., 2022) on the augmented data as a warm-up or TAPT (Task-Adaptive-Pre-Training) and then continued the fine-tuning on the provided SemRel data.

NRK They ensembled various BERT-like models and used a weighted voting technique to improve the performance of their model.

PEAR They examined the effect of combining or using per-language data through 5-fold validation. They did not conduct any text preprocessing to maintain fairness across languages. They defined three model configurations: "base" with no training, "all" trained on all languages, and "lan" trained on one language. They experimented with multilingual embeddings, cross-encoders, and augmented data from bi-encoders.

5.4.2 Popular Methods

The general trend for the methods submitted to track A was (1) embedding sentence pairs into text and (2) training a regression model. Some teams used traditional embeddings and regression approaches (e.g., word2vec with support vector regressor – team 'Text Mining'). The majority used deep learning approaches (e.g., BERT, RoBERTa) or other large pre-trained transformer models (e.g., teams "IITK", "Fired_from_NLP, HausaNLP"). When using these models, the teams would often experiment with different hyperparameters. Some teams went further and modified the specific learning approach or representations learned through methods such as contrastive learning (e.g., team: IITK).

5.4.3 Most Effective and Original Methods

In track A, the participants used the provided training sets for each of the 9 languages included in the track (`amh`, `arq`, `ary`, `eng`, `esp`, `hau`, `kin`, `mar` and `tel`). Overall, the different teams explored several approaches to enhance the performance. For instance, the top performing team PALI, used MT-DNN (Multi-Task Deep Neural Networks for Natural Language Understanding) (Liu et al., 2019a) and outperformed all the other teams across all languages except for Spanish and Kinyarwanda. For Kinyarwanda, king001 who used MT for data augmentation and multilingual mixed training and XLM-R (Conneau et al., 2020) as a base model achieved the best performance, and AAdaM who used translation-based data augmentation and adapter-based tuning reported the best score.

Note. however, that since PALI and king001 did not submit system description papers, they are not ranked in Tables 3 and 4.

5.5 Task B: Unsupervised

5.5.1 Best Performing Systems

SATLab Team SATLab used a system based on a model developed for authorship identification of source code (Bestgen, 2019). The system processed each pair of utterances independently, generating a distance between them without relying on additional information. Their pre-processing involved lower-casing of texts and making use of character n-grams ranging from 1 to 5 characters, encompassing all characters including spaces, punctuation marks, symbols, and characters from different writing systems. All n-grams were retained without a frequency threshold. The frequency of each feature was weighted by a logarithmic function, and the features of each statement were weighted by the L2 norm. The semantic similarity between utterances was estimated using the Euclidean distance between sets of n-grams in each pair.

MasonTigers In the initial phase, team MasonTigers obtained the embeddings of training data instances and used TF–IDF, PPMI, LaBSE sentence transformer, and language-specific BERT models for multiple languages. Cosine similarity scores were then computed between pairs of embeddings, followed by the use of ElasticNet and Linear Regression separately to predict sentence pair similarity. Predicted values were clipped to ensure a range from 0 to 1.

HW–TSC Team HW-TSC's method included the N-gram chars utilising tokenizers from XLM-RoBERTa and m-BERT as key features to compute similarity scores based on n-gram dictionaries of sentences. They also used BERTScore to assess text quality based on the cosine similarity of token-level representations from the BERT model.

5.5.2 Popular Methods

As the main challenge with track B was the prevention of using any data of more than two words long related to semantics, many teams such as Hausa–NLP and Tübingen–CL used pre-trained language models such as All-MiniLM-L6-v2 (Reimers and Gurevych, 2019).

Most teams opted for language-specific data and models, if not trained on similarity data, and compared the performance to monolingual BERT models. However, none of these methods were used by the top three performing teams.

5.5.3 Most effective and Original Methods

The most effective methods for the unsupervised track for all languages were submitted by teams SATLab, MasonTigers, and HW–TSC (top–3). SATLab's approach involved processing pairs independently using character n-grams. MasonTigers, on the other hand, leveraged various embedding methods and statistical machine learning using simple features such as TF-IDF and BERT models to compute the cosine similarity between embeddings, further refined using ElasticNet. On the other hand, The HW–TSC team used innovative techniques

Rank	Team	afr	amh	arb	arq	ary	eng	esp	hau	hin	ind	kin	pun	Average
1	SATLab (Bestgen, 2024)	0.761	**0.764**	0.487	**0.521**	**0.599**	0.774	**0.709**	**0.513**	0.649	**0.491**	0.458	-0.215	**0.543**
2	MasonTigers (Goswami et al., 2024)	0.757	0.656	0.405	0.424	0.561	0.766	0.661	0.504	0.571	0.382	**0.465**	0.020	0.514
3	HW–TSC (Piao et al., 2024)	0.639	0.650	0.402	0.296	0.460	0.758	0.641	0.382	0.613	0.445	0.323	**0.173**	0.482
4	UAlberta (Shi et al., 2024)	**0.789**	0.723	0.467	0.368	0.063	0.775	0.680	0.380	0.691	0.484	0.378	-0.027	0.481
*	Lexical Overlap	0.706	0.633	0.320	0.400	0.627	0.670	0.670	0.306	0.527	0.553	0.333	-0.274	0.456
5	silp_nlp (Singh et al., 2024)	0.732	0.643	0.314	0.402	0.552	0.317	-	0.387	0.571	0.532	0.350	-0.110	0.400
6	HausaNLP (Salahudeen et al., 2024)	0.716	0.038	0.202	0.334	0.397	0.819	0.618	0.358	0.440	0.407	0.404	-0.084	0.387
*	SemRel baseline (XLMR)	0.562	0.573	0.316	0.247	0.174	0.601	0.689	0.041	0.507	0.467	0.132	-0.072	0.353
NR	IITK (Basak et al., 2024)	-	0.068	-	0.489	0.358	0.808	0.591	0.379	-	-	-	-	0.449
NR	YSP (Aali et al., 2024)	-	-	-	0.385	-	0.788	0.598	0.193	-	-	0.377	-	0.468
NR	Tübingen-CL (Zhang and Çöltekin, 2024)	-	-	-	-	-	**0.837**	0.705	-	0.649	-	-	-	0.730
NR	CAILMD-23 (Sonavane et al., 2024)	-	-	-	-	-	0.819	-	-	**0.797**	-	-	-	0.808
NR	Self-StrAE (Opper and Narayanaswamy, 2024)	0.765	-	-	-	-	-	0.635	-	-	-	-	-	0.700
NR	NLU-STR (Malaysha et al., 2024)	-	-	**0.489**	-	-	-	-	-	-	-	-	-	0.489

Table 5: Track B results. The best results are in bold, and NR stands for *not ranked*. As the methods are highly language-dependent, we only rank teams that participated in at least 8 sub-tracks, but we highlight in blue the best results achieved by non-ranked teams. (Non-ranked teams are sorted based on the number of languages they participated in.)

such as the N-gram chars method with XLM-R and m-BERT tokenizers, as well as the BERTScore to evaluate the text quality.

In Table 5, we also have honorable mentions for teams that did not participate in all the languages but achieved remarkable results in one or a few languages. Notably, team CAILMD–23 achieved the best results in Hindi by using Hindi-BERT-v2, and team Tübingen–CL achieved the best results in English.

5.6 Task C: Crosslingual

5.6.1 Best Performing Systems

AAdaM They experimented with full fine-tuning, adapter fine-tuning using MAD (Pfeiffer et al., 2020), and data augmentation using different language combinations to augment data in a given source language.

UAlberta They used an XGBoost regressor-based (Chen and Guestrin, 2016) ensemble approach to integrate the predicted relatedness scores of three distinct regression models, with one optional SBERT model, as input and returned the final relatedness score as output. They applied the English version of their method trained for Track A to the translations of the non-English test sets. The regression model fine-tuned on MPNet was used in the XGBoost ensemble only for `amh`, `hau`, and `hin`, but not for the other languages such as `esp`, `ary`, `kin`, `ind`, `arb`, `arq`, and `afr`. The pre-trained English language models that were used include RoBERTa Large, T5 Base, and GPT2 Base, as well as MPNet only for languages `amh`, `hau`, and `hin`.

silp_nlp They used the provided datasets and cross-lingual transferability with all the provided datasets, except data in the target language, as a source. Their cross-lingual transfer approach made use of MuRIL (Khanuja et al., 2021) which led to the best results for Hindi and XLM-R (Conneau et al., 2020) led to the best ones for all the other languages.

5.6.2 Popular Methods

For the crosslingual track, many teams including best-performing ones such as UAlberta chose approaches similar to the ones used for supervised sub-tracks (e.g., using an XGBoost regressor (Chen and Guestrin, 2016)). As the main challenge was to determine how to leverage data in languages other than the target, many teams combined the provided SemRel datasets in all possible languages (e.g., king001, AAdaM). Some used the training datasets without any modifications (e.g., team HausaNLP) and others experimented with different language combinations to use those that would lead to the best results (e.g., MasonTigers). Finally, some teams applied advanced techniques to modify the vector embedding space (e.g., by adjusting for the anisotropic nature of vector spaces – team: USTC-CTSU).

5.6.3 Most Effective and and Original Methods

Overall, applying methods that are similar to the ones used in the supervised track using data in different languages can indeed lead to good results (e.g., king001, AAdaM, UAlberta). In addition, combining data in different languages and testing on another could boost the performance of crosslin-

Rank	Team	afr	amh	arb	arq	ary	eng	esp	hau	hin	ind	kin	pun	Average
1	AAdaM (Zhang et al., 2024)	0.814	0.863	0.653	0.551	0.600	0.794	0.621	0.729	0.839	**0.528**	0.650	**0.155**	**0.650**
2	UAlberta (Shi et al., 2024)	0.806	0.816	**0.671**	0.441	0.602	-	0.572	0.678	0.828	0.449	0.636	-0.017	0.589
*	SemRel baseline (LaBSE)	0.786	0.838	0.615	0.463	0.404	0.800	0.623	0.625	0.760	0.472	0.571	-0.049	0.579
3	silp_nlp (Singh et al., 2024)	0.747	0.805	0.427	0.387	0.673	0.737	0.569	0.643	0.801	0.472	-	-0.037	0.566
4	MaiNLP (Zhou et al., 2024)	0.738	0.728	0.399	0.274	0.568	-	-	-	0.695	0.319	**0.681**	0.087	0.499
5	USTCCTSU (Li et al., 2024a)	0.603	0.656	0.469	0.420	0.402	0.700	0.689	0.111	0.596	0.476	0.302	-0.084	0.445
6	umbclu (Roy Dipta and Vallurupalli, 2024)	**0.822**	0.043	0.035	0.126	-0.038	0.788	0.609	0.457	0.155	0.515	0.484	-0.078	0.326
7	HausaNLP (Salahudeen et al., 2024)	0.737	-0.031	0.184	0.074	0.276	0.360	0.604	0.177	0.346	0.472	0.319	0.114	0.303
8	MasonTigers (Goswami et al., 2024)	0.385	0.131	0.213	0.221	0.203	0.310	0.557	0.099	0.511	0.133	0.079	0.020	0.239
NR	USTC_NLP	0.749	0.709	0.517	0.414	0.613	0.784	0.685	0.476	0.658	0.460	0.454	-0.248	0.523
NR	king001	0.810	**0.878**	0.657	**0.614**	**0.820**	-	**0.708**	**0.733**	**0.844**	0.376	0.630	-0.050	0.641
NR	saturn	0.818	0.814	-	-	-	-	-	0.569	-	-	0.604	-0.103	0.540
NR	YSP (Aali et al., 2024)	-	-	-	0.225	-	**0.819**	0.657	0.212	-	-	0.256	-	0.434
NR	CAILMD-23 (Sonavane et al., 2024)	-	-	-	-	-	0.786	-	-	0.810	-	-	-	0.798
NR	PALI	-	-	-	-	0.842	-	-	-	-	-	-	-	0.842
NR	faridlazuarda	-	-	-	-	-	-	-	-	-	-	0.600	0.058	0.329
NR	ETMS@IITKGP	-	-	-	-	-	-	0.549	-	-	-	-	-	0.549
NR	Silp_nlp	-	-	-	-	-	-	-	-	-	0.472	-	-	0.472
NR	lukmanaj	-	-	-	-	-	-	-	0.177	-	-	-	-	0.177

Table 6: Track C results. The best results are in bold, and NR stands for *not ranked*. As the methods are highly language-dependent, we only rank teams that participated in at least 8 sub-tracks, but we highlight in blue the best results achieved by non-ranked teams. (Non-ranked teams are sorted based on the number of languages they participated in.)

gual models for STR as shown by team sil_nlp who achieved the best results in Amharic and Moroccan Arabic. Further, we note that leveraging advanced features such as (1) linguistic features (e.g., language family) as performed by MaiNLP, who achieved the best results for Kinyarwanda, and (2) embedding features by adjusting the distribution of the similarity scores as experimented by USTCCTSU could also help boost the performance.

Besides reporting on the best-performing teams only, in Table 6, we also mention teams that did not participate in many sub-tracks but achieved good results such as team YSP, which outperforms all the other teams in English.

6 Discussion

We observe that in general, teams opt out of pretrained models, and in most cases, the methods do not perform equally well across languages. Hence, for a given track, performing well in a language does not mean performing equally well in another language.

Further, the results show that good scores are not only related to low vs. high-resourcedness. For instance, In tracks B and C, results for Modern Standard Arabic (arb), which is considered high resource, are sometimes worse than those for low resource languages such as Amharic (amh) and Kinyarwanda (kin).

Interestingly, although the participating teams rarely use language-specific features, such approaches lead to good and interpretable results, as reported by e.g., team MaiNLP, who leveraged information about language families in Track C. We also note that for Track C, using a simple LaBSE baseline can achieve results that are better or comparable to more sophisticated techniques (see Ousidhoum et al. (2024) for language-specific baseline results).

7 Conclusion

We presented the first shared task on semantic relatedness, covering three tracks and 14 languages in total. The submitted systems were ranked based on the ranking of their predicted relatedness scores compared to the gold labels.

We summarised the reported results, the best-performing methods, and the most effective, promising, and original ones. Overall, our findings on sentence representation techniques vary across the different languages and show that determining semantic textual relatedness is not a trivial task.

8 Limitations

As stated in Ousidhoum et al. (2024), we acknowledge that there is no formal definition of what constitutes semantic relatedness and that our annotations may be subjective. To mitigate the issue, we share our guidelines and annotated instances so researchers in the community can expand on our work, replicate it, and study the disagreements in our data. We are also aware of the limited number of data sources and data variety in some low-resource languages involved. We do not claim

that the datasets released represent all variations of these languages. However, they remain a good starting point as they were carefully picked, labeled, and processed by native speakers.

9 Ethics Statement

As stated in Ousidhoum et al. (2024), we acknowledge all the possible socio-cultural biases that can come with our data due to the data sources or the annotation process. When building our datasets, we did avoid instances with inappropriate or offensive utterances, but we might have missed some. Our goal was to identify common perceptions of semantic relatedness by native speakers, and our labels are not meant to be standardised for any given language as these are not fully representative of its usage.

References

Yasamin Aali, Sardar Hamidian, and Parsa Farinneya. 2024. Ysp at semeval-2024 task 1: Enhancing sentence relatedness assessment using siamese networks. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 946–950, Mexico City, Mexico. Association for Computational Linguistics.

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-lingual Evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Carlos Santos Armendariz, Matthew Purver, Senja Pollak, Nikola Ljubešić, Matej Ulčar, Ivan Vulić, and Mohammad Taher Pilehvar. 2020. SemEval-2020 task 3: Graded word similarity in context. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 36–49, Barcelona (online). International Committee for Computational Linguistics.

Senthil Kumar B, Aravindan Chandrabose, Gokulakrishnan B, and Karthikraja TP. 2024. NLP_Team1@SSN at semeval-2024 task 1: Impact of language models in sentence-bert for semantic textual relatedness in low-resource languages. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1866–1871, Mexico City, Mexico. Association for Computational Linguistics.

Udvas Basak, Rajarshi Dutta, Shivam Pandey, and Ashutosh Modi. 2024. IITK at semeval-2024 task 1: Contrastive learning and autoencoders for semantic textual relatedness in multilingual texts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1454–1459, Mexico City, Mexico. Association for Computational Linguistics.

Abdessamad Benlahbib, Anass Fahfouh, Hamza Alami, and Achraf Boumhidi. 2024. NLP-LISAC at semeval-2024 task 1: Transformer-based approaches for determining semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 213–217, Mexico City, Mexico. Association for Computational Linguistics.

Yves Bestgen. 2019. CECL at SemEval-2019 task 3: Using surface learning for detecting emotion in textual conversations. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 148–152, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yves Bestgen. 2024. Satlab at semeval-2024 task 1: A fully instance-specific approach for semantic textual relatedness prediction. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 95–100, Mexico City, Mexico. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Lee J Cronbach. 1951. Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3):297–334.

Fahad Ebrahim and Mike Joy. 2024. WarwickNLP at semeval-2024 task 1: Low-rank cross-encoders for efficient semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 246–252, Mexico City, Mexico. Association for Computational Linguistics.

Seyedeh Fatemeh Ebrahimi, Karim Akhavan Azari, Amirmasoud Iravani, Hadi Alizadeh, Zeinab Taghavi, and Hossein Sameti. 2024. Sharif-STR at semeval-2024 task 1: Transformer as a regression model for fine-grained scoring of textual semantic relations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1032–1041, Mexico City, Mexico. Association for Computational Linguistics.

Alex Eponon and Luis Ramos Perez. 2024. Pinealai at semeval-2024 task 1: Exploring semantic relatedness prediction using syntactic, tf-idf, and distance-based features. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 922–926, Mexico City, Mexico. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

T. N. Flynn and A. A. J. Marley. 2014. Best-worst scaling: theory and methods. In Stephane Hess and Andrew Daly, editors, *Handbook of Choice Modelling*, pages 178–201. Edward Elgar Publishing.

Shreejith G, Ravindran V, Aashika Jetti, Rajalakshmi Sivanaiah, and Angel Deborah S. 2024. Techssn at semeval-2024 task 1: Multilingual analysis for semantic textual relatedness using boosted transformer models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 894–899, Mexico City, Mexico. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dhiman Goswami, Sadiya Sayara Chowdhury Puspo, Md Nishat Raihan, Al Nahian Bin Emran, Amrita Ganguly, and Marcos Zampieri. 2024. Masontigers at semeval-2024 task 1: An ensemble approach for semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1370–1380, Mexico City, Mexico. Association for Computational Linguistics.

Ruqaiya Hasan and Michael AK Halliday. 1976. Cohesion in english. *London, 1976; Martin JR*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Md. Sajjad Hossain, Ashraful Islam Paran, Symom Hossain Shohan, Jawad Hossain, and Mohammed Moshiul Hoque. 2024. SemanticCUET-Sync at semeval-2024 task 1: Finetuning sentence transformer to find semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1212–1218, Mexico City, Mexico. Association for Computational Linguistics.

Tollef Jørgensen. 2024. PEAR at semeval-2024 task 1: Pair encoding with augmented re-sampling for semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1395–1401, Mexico City, Mexico. Association for Computational Linguistics.

Ron Keinan. 2024. Text mining at semeval-2024 task 1: Evaluating semantic textual relatedness in low-resource languages using various embedding methods and machine learning regression models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 407–418,

Mexico City, Mexico. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha P. Talukdar. 2021. Muril: Multilingual representations for indian languages. *CoRR*, abs/2103.10730.

Svetlana Kiritchenko and Saif M. Mohammad. 2016. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, San Diego, California.

Svetlana Kiritchenko and Saif M Mohammad. 2017a. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. *arXiv preprint arXiv:1712.01765*.

Svetlana Kiritchenko and Saif M. Mohammad. 2017b. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of The Annual Meeting of the Association for Computational Linguistics (ACL)*, Vancouver, Canada.

G Frederic Kuder and Marion W Richardson. 1937. The theory of the estimation of test reliability. *Psychometrika*, 2(3):151–160.

Jianjian Li, Shengwei Liang, Yong Liao, Hongping Deng, and Haiyang Yu. 2024a. USTCCTSU at semeval-2024 task 1: Reducing anisotropy for cross-lingual semantic textual relatedness task. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 868–874, Mexico City, Mexico. Association for Computational Linguistics.

Weijie Li, Jin Wang, and Xuejie Zhang. 2024b. Ynu-hpcc at semeval-2024 task 1: Self-instruction learning with black-box optimization for semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 779–786, Mexico City, Mexico. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Jordan J Louviere and George G Woodworth. 1991. Best-Worst Scaling: A Model For The Largest Difference Judgments. Technical report, Working paper.

Anand Kumar M and Hemanth Kumar M. 2024. scalar semeval-2024 task 1: Semantic textual relatednes for english. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 889–893, Mexico City, Mexico. Association for Computational Linguistics.

Sanad Malaysha, Mustafa Jarrar, and Mohammed Khalilia. 2024. NLU-STR at semeval-2024 task 1: Generative-based augmentation and encoder-based scoring for semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 881–888, Mexico City, Mexico. Association for Computational Linguistics.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Daniela Moctezuma, Eric Tellez, and Mario Graff. 2024. Ingeotec at semeval-2024 task 1: Bag of words and transformers. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1144–1148, Mexico City, Mexico. Association for Computational Linguistics.

Saif M Mohammad and Graeme Hirst. 2012. Distributional Measures of Semantic Distance: A Survey. *arXiv preprint arXiv:1203.1858*.

Anderson Morillo, Daniel Peña, Juan Carlos Martinez Santos, and Edwin Puertas. 2024. Verbanexai lab at semeval-2024 task 1: A multilayer artificial intelligence model for semantic relationship detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1334–1340, Mexico City, Mexico. Association for Computational Linguistics.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics*, 17(1):21–48.

Kiet Nguyen and Dang Thin. 2024. NRK at semeval-2024 task 1: Semantic textual relatedness through domain adaptation and ensemble learning on bert-based models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 76–81, Mexico City, Mexico. Association for Computational Linguistics.

Mattia Opper and Siddharth Narayanaswamy. 2024. Self-StrAE at semeval-2024 task 1: Making self-structuring autoencoders learn more with less. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 108–115, Mexico City, Mexico. Association for Computational Linguistics.

Bryan Orme. 2009. Maxdiff analysis: Simple counting, individual-level logit, and HB. Sawtooth Software, Inc.

Jesus-German Ortiz-Barajas, Gemma Bel-Enguix, and Helena Goméz-Adorno. 2024. MBZUAI-UNAM at semeval-2024 task 1: Sentence-crobi, a simple cross-bi-encoder-based neural network architecture for semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1060–1068, Mexico City, Mexico. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, Meriem Beloucif, Chris Biemann, Sofia Bourhim, Christine De Kock, Genet Shanko Dekebo, Oumaima Hourrane, Gopichand Kanumolu, Lokesh Madasu, Samuel Rutunda, Manish Shrivastava, Thamar Solorio, Nirmal Surange, Hailegnaw Getaneh Tilaye, Krishnapriya Vishnubhotla, Genta Winata, Seid Muhie Yimam, and Saif M. Mohammad. 2024. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Mengyao Piao, Su Chang, Yuang Li, Xiaosong Qiao, Xiaofeng Zhao, Yinglu Li, Min Zhang, and Hao Yang. 2024. Hw-tsc 2024 submission for the semeval-2024 task 1: Semantic textual relatedness (str). In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1645–1649, Mexico City, Mexico. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shubhashis Roy Dipta and Sai Vallurupalli. 2024. UMBCLU at semeval-2024 task 1: Semantic textual relatedness with and without machine translation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1341–1347, Mexico City, Mexico. Association for Computational Linguistics.

Saheed Abdullahi Salahudeen, Falalu Ibrahim Lawan, Yusuf Aliyu, Amina Abubakar, Lukman Aliyu, Nur Bala Rabiu, Mahmoud Said Ahmad, Idi Mohammed, Aliyu Rabiu Shuaibu, Alamin Musa, Auwal Shehu Ali, and Zedong Nie. 2024. Hausanlp at semeval-2024 task 1: Textual relatedness analysis for semantic representation of sentences. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 188–192, Mexico City, Mexico. Association for Computational Linguistics.

Anik Shanto, Md. Sajid Alam Chowdhury, Mostak Chowdhury, Udoy Das, and Hasan Murad. 2024. Fired_from_NLP at semeval-2024 task 1: Towards developing semantic textual relatedness predictor: A transformer-based approach. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 846–851, Mexico City, Mexico. Association for Computational Linguistics.

Ning Shi, Senyu Li, Guoqing Luo, Amirreza Mirzaei, Ali Rafiei, Jai Riley, Hadi Sheikhi, Mahvash Siavashpour, Mohammad Tavakoli, Bradley Hauer, and Grzegorz Kondrak. 2024. UAlberta at semeval-2024 task 1: A potpourri of methods for quantifying multilingual semantic textual relatedness and similarity. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1810–1817, Mexico City, Mexico. Association for Computational Linguistics.

Marco Siino. 2024. All-mpnet at semeval-2024 task 1: Application of mpnet for evaluating semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 372–377, Mexico City, Mexico. Association for Computational Linguistics.

Sumit Singh, Pankaj Kumar Goyal, and Uma Shanker Tiwary. 2024. silp_nlp at semeval-2024 task 1: Cross-lingual knowledge transfer for mono-lingual learning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1187–1193, Mexico City, Mexico. Association for Computational Linguistics.

Srushti Sonavane, Sharvi Endait, Ridhima Sinare, Pritika Rohera, Advait Naik, and Dipali Kadam. 2024. Cailmd-23 at semeval-2024 task 1: Multilingual evaluation of semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 969–974, Mexico City, Mexico. Association for Computational Linguistics.

Lianshuang Su and Xiaobing Zhou. 2024. Nlp_str_teams at semeval-2024 task1: Semantic textual relatedness based on mask prediction and bert model. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 330–334, Mexico City, Mexico. Association for Computational Linguistics.

Hidetsune Takahashi, Xingru Lu, Sean Ishijima, Deokgyu Seo, Yongju Kim, Sehoon Park, Min Song, Kathylene Marante, Keitaro-Luke Iso, Hirotaka Tokura, and Emily Ohman. 2024. OZemi at semeval-2024 task 1: A simplistic approach to textual relatedness evaluation using transformers and machine translation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*,

pages 7–12, Mexico City, Mexico. Association for Computational Linguistics.

Dilip Venkatesh and Sundaresan Raman. 2024. Bits pilani at semeval-2024 task 1: Using text-embedding-3-large and labse embeddings for semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 852–855, Mexico City, Mexico. Association for Computational Linguistics.

Bin Wang, C.-C. Jay Kuo, and Haizhou Li. 2022. Just rank: Rethinking evaluation with word and sentence similarities. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6060–6077, Dublin, Ireland. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leixin Zhang and Çağrı Çöltekin. 2024. Tübingen-CL at semeval-2024 task 1: Ensemble learning for semantic relatedness estimation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1008–1014, Mexico City, Mexico. Association for Computational Linguistics.

Miaoran Zhang, Mingyang Wang, Jesujoba Alabi, and Dietrich Klakow. 2024. Aadam at semeval-2024 task 1: Augmentation and adaptation for multilingual semantic textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 787–797, Mexico City, Mexico. Association for Computational Linguistics.

Shijia Zhou, Huangyan Shan, Barbara Plank, and Robert Litschko. 2024. MaiNLP at semeval-2024 task 1: Analyzing source language selection in cross-lingual textual relatedness. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1854–1865, Mexico City, Mexico. Association for Computational Linguistics.

A Appendix: Track A–Best Performing Teams

PALI and king001 Both teams PALI and king001 did not submit a task description paper. king001 chose to use translation for data augmentation and multilingual mixed training. The team used XLM–R as their base model and DeBERTa–v3 (He et al., 2021).

AAdaM Team AAdaM opted for translation-based data augmentation to increase the training data size for better performance. The English STR training data and STSB (semantic similarity) data were translated to create augmented datasets in other languages. The team explored both fine-tuning and adapter-based tuning, aiming to examine and compare their effectiveness on STR across the different languages. Given a target language, they first fine-tuned the cross-encoder-based AfroXLMR model on the augmented data as a warm-up or TAPT (Task-Adaptive-Pre-Training) and then continued the fine-tuning on the provided STR data.

NRK They used ensembling and various BERT-like models.

PEAR They examined the effect of combining vs. using language-specific data through 5-fold validation. No text preprocessing was conducted to maintain fairness across languages. Three model configurations were defined: “base” with no training, “all” trained on all languages, and “lang” trained on one language. They experimented with multilingual embeddings, cross-encoders, and data augmentation with bi-encoders. Parameter optimization was conducted using Optuna.

silp_nlp Team silp_nlp’s methodology for track A was a two-stage training. In the initial stage, they trained a model using all 9 languages covered in track A with MuRIL (Khanuja et al., 2021). They experimented with different hyperparameters on five epochs and selected the best multilingual checkpoint based on the average validation data loss. They fine-tuned the resulting model using the training data for each language in track A and ended up with monolingual models.

Each monolingual model was trained using different hyperparameters and they selected their final model based on the validation data loss of the corresponding language track.

NLP_1@SSN They used SBERT fine-tuned on multilingual and monolingual pre-trained language models Overall, they observed that the usage of monolingual PLMs did not guarantee better performance.

UAlberta They used an ensemble approach with an XGBoost regression (Chen and Guestrin, 2016) to integrate the predicted relatedness scores of three distinct regression models, with one optional SBERT model, as input and returned the final relatedness scores as output. Each of these models used a different pre-trained language model as its backbone, specifically RoBERTa Large (Liu et al.,

2019b), T5 Base, GPT-2 Base, and the optional SBERT (MPNet). They merged the English training and development sets with the translated training set of the target language. Then, they split them again via uniform random sampling according to their original sizes to establish new training and development splits. The did not use the data provided for arq, ary, and kin, and applied the English-trained version of their method to the English translations of the arq, ary, and kin test sets instead.

MBZUAI-UNAM They fine-tuned a paraphrase model architecture to train language-specific models, using a separate pre-trained model to embed each language. They also experimented with combined training sets based on the language families.

INGEOTEC For English and Spanish, they used embeddings (microsoft/mpnet-base, bert-base-multilingual-cased) to train an SVM classifier. For the other languages, they used prior work EvoMSA.

HausaNLP They used different base pre-trained models.

B Appendix: Track B

SATLab They proposed a system based on a model developed for the authorship identification of source code (Bestgen, 2019). It processed each pair of utterances independently, generating a distance between them without relying on additional information. Pre-processing involved lower-casing of texts. Character n-grams ranging from 1 to 5 characters are used, encompassing all characters including spaces, punctuation marks, symbols, and characters from different writing systems, all n-grams are retained without a frequency threshold. The frequency of each feature was weighted by a logarithmic function, and the features of each statement were weighted by the L2 norm. Semantic similarity between utterances was estimated using Euclidean distance between sets of n-grams in each pair.

MasonTigers In the initial phase, team MasonTigers obtained embeddings of training data and used various methods including TF-IDF, PPMI, LaBSE sentence transformer, and language-specific BERT models for multiple languages. Cosine similarity was then computed between pairs of embeddings, followed by applying ElasticNet and Linear Regression separately to predict sentence pair similarity in the development phase. Predicted values were clipped to ensure a range from 0 to 1.

HW–TSC The key features used by team HW-TSC's method included the N-gram chars method using XLM-RoBERTa and m-BERT tokenizers to compute similarity scores based on n-gram sentence dictionaries. They also used the BERTScore method to assess text quality based on the cosine similarity of token-level representations from the BERT model.

UAlberta They used a linear combination of two sets of normalized results, each derived from the cosine similarity measurements of sentence embeddings obtained from the hidden sentence representations processed by BERT Large and RoBERTa Large. They calculated the final relatedness scores by averaging the cosine similarity scores of sentence embeddings obtained from each set.

silp_nlp They converted the sentences into unigram and bigram representations and used Support Vector Regression (SVR).

Sentences were combined and transformed into a vector, and each sentence was indexed based on a value that represented the count of unigrams/bigrams present in it. The resulting vector was fed into the SVR model along with label values for training.

HausaNLP Team HausaNLP used a standard all-MiniLM-L6-v2 model to train a model for Track B.

IITK Team IITK uses SimCSE (Gao et al., 2021), or Simple Contrastive Learning of Sentence Embeddings that induced slight variations in its representation through dropout. TSDAE(Wang et al., 2021), a denoising autoencoder, was used to generate sentence embeddings by reconstructing original sentences in the presence of noise. They used BERT to construct the denoising autoencoder and TSDAE optimized the likelihood of reconstructing sentences during training, which led to compact embeddings.

Tübingen-CL Team Tübingen-CL opted for exploring features like cosine distance of average word embeddings and word overlap ratios, to potentially enhance performance. For English, they used two models: multi-qa-MiniLM-L6-cos-v1 trained on QA pairs and trained for semantic search and e5-

base-unsupervised trained on various pairs including question-answer and post-comment pairs, both refined with unsupervised transformation (PCA). Two additional features, PCA-transformed GloVe embeddings, and content word overlap ratios were incorporated into the unsupervised ensemble system. Similar methods were applied for Spanish and Hind using multilingual BERT embeddings and various feature combinations to predict relatedness.

CAILMD-23 Team CAILMD-23 participated in the English and Hindi sub-tracks of the unsupervised task. They experimented with a few models such as BERT-based and Hindi-Bert v2. The latter is trained on Hindi text comprehension with a training corpus of roughly 1.8 billion tokens.

C Appendix: Track C

AAdaM hey experimented with full fine-tuning, adapter fine-tuning using MAD (Pfeiffer et al., 2020), and data augmentation using different language combinations to augment data in a given source language.

king001 They did not submit a system description paper but they reported combining the training datasets provided for track A, and if one of them was in the target language, they translated it into English. Then, they run multi-task learning for 15 epochs.

UAlberta They used an ensemble approach with an XGBoost regressor (Chen and Guestrin, 2016) to integrate the predicted relatedness scores of three distinct regression models, with one optional SBERT model, as input. Each of their models used a different pre-trained language model as its backbone, specifically RoBERTa Large, T5 Base, GPT-2 Base, and the optional SBERT (MPNet).

They applied the English version of their method reported for Track A to the translations of the non-English test sets. The regression model fine-tuned on MPNet was used in the XGBoost ensembling method for amh, hau, and hin and not for esp, ary, kin, ind, arb, arq, and afr.

silp_nlp They used cross-lingual transferability on all the provided datasets except for the target language (e.g., when they test on Telugu, they use all languages except Telugu). In their cross-lingual transfer approach, MuRIL (Khanuja et al., 2021) led to the best results for Hindi and XLM-R (Conneau et al., 2020) for all the other languages.

USTCCTSU They used XLM-R (Conneau et al., 2020) trained on a combination of language inputs (chosen by trying different combinations with the best one including all the languages). They ranked in the top 5 for ind,arq, and esp.

They adjusted the similarity scores for the XLM-R base models by applying a technique called *whitening* that allowed them to change the non-uniform score distribution into multiple distributions, and eventually, into a uniform one.

MaiNLP They finetuned multilingual LLMs (XLM-R and Furina) using an upscaled version of the data from Track A. They assessed the linguistic similarity of the available Track A data to determine the most useful datasets and experimented with different language families. For pre-processing, they used tokenization, segmentation, and translation. They also experimented with transliteration to change the scripts into Latin. Translations helped them upscale the English, Hausa, and Spanish training data and then evaluate on the Track C data. They achieved the best results for Kinyarwanda.

umbclu They pre-trained T5 models with SemRel data. They used the English fine-tuned models for inference on all language test sets except English. On the other hand, they used Spanish models for inference on English.

HausaNLP They used a BERT-based model fine-tuned on the datasets in other languages. E.g., they trained on English data and tested on Spanish, trained on Kinyarwanda and tested on Hausa. They ranked in the top 5 in Task C for ind, pan.

MasonTigers They used statistical machine learning (Linear Regression, ElasticNet with TF-IDF and PPMI features) along with language-specific BERT-based models to predict the relatedness scores. The models were trained on dataset combinations of 5 languages other than the target language and used BERT-based models's similarity prediction on the target test data (e.g., they trained on amh, eng, esp, arq, ary and tested on afr). For language-specific BERT-like models, they used African language BERT-base models, Arabic BERT-based models, AfricanBERTa, and for eng, hin, ind, pun, esp, they used spanBERTa, BanglaBERT, RoBERTa-tagalog-base-BERT, HindiBERT, and RoBERTa.

Team	Paper
AAdaM	Zhang et al. (2024)
All–Mpnet	Siino (2024)
BITS Pilani	Venkatesh and Raman (2024)
CAILMD–23	Sonavane et al. (2024)
Fired_from_NLP	Shanto et al. (2024)
HausaNLP	Salahudeen et al. (2024)
HW-TSC	Piao et al. (2024)
IITK	Basak et al. (2024)
INGEOTEC	Moctezuma et al. (2024)
MaiNLP	Zhou et al. (2024)
MasonTigers	Goswami et al. (2024)
MBZUAI–UNAM	Ortiz-Barajas et al. (2024)
NLP–LISAC	Benlahbib et al. (2024)
NLP_STR_teamS	Su and Zhou (2024)
NLP_Team1SSN	B et al. (2024)
NLU–STR	Malaysha et al. (2024)
NRK	Nguyen and Thin (2024)
OZemi	Takahashi et al. (2024)
PEAR	Jørgensen (2024)
Pinealai	Eponon and Ramos Perez (2024)
SATLab	Bestgen (2024)
scaLAR	M and M (2024)
Self–StrAE	Opper and Narayanaswamy (2024)
SemanticCUETSync	Hossain et al. (2024)
Sharif_STR	Ebrahimi et al. (2024)
silp_nlp	Singh et al. (2024)
TECHSSN	G et al. (2024)
Text Mining	Keinan (2024)
Tübingen–CL	Zhang and Çöltekin (2024)
UAlberta	Shi et al. (2024)
UMBCLU	Roy Dipta and Vallurupalli (2024)
USTCCTSU	Li et al. (2024a)
VerbaNexAI	Morillo et al. (2024)
WarwickNLP	Ebrahim and Joy (2024)
YNU–HPCC	Li et al. (2024b)
YSP	Aali et al. (2024)

Table 7: The participating teams (alphabetically ordered) that submitted system description papers.

SemEval-2024 Task 6: SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes

Timothee Mickus **Elaine Zosa** **Raúl Vázquez** **Teemu Vahtola**
Jörg Tiedemann **Vincent Segonne** **Alessandro Raganato** **Marianna Apidianaki**
University of Helsinki Silo AI, Finland Université Bretagne Sud
University of Milano-Bicocca University of Pennsylvania
{firstname.lastname}@{helsinki.fi, silo.ai, univ-ubs.fr, unimib.it}
marapi@seas.upenn.edu

Abstract

This paper presents the results of the SHROOM, a shared task focused on detecting hallucinations: outputs from natural language generation (NLG) systems that are fluent, yet inaccurate. Such cases of overgeneration put in jeopardy many NLG applications, where correctness is often mission-critical. The shared task was conducted with a newly constructed dataset of 4000 model outputs labeled by 5 annotators each, spanning 3 NLP tasks: machine translation, paraphrase generation and definition modeling.

The shared task was tackled by a total of 58 different users grouped in 42 teams, out of which 26 elected to write a system description paper; collectively, they submitted over 300 prediction sets on both tracks of the shared task. We observe a number of key trends in how this approach was tackled—many participants rely on a handful of model, and often rely either on synthetic data for fine-tuning or zero-shot prompting strategies. While a majority of the teams did outperform our proposed baseline system, the performances of top-scoring systems are still consistent with a random handling of the more challenging items.

1 Introduction

The modern NLG landscape is plagued by two interlinked problems: On the one hand, our current neural models have a propensity to produce inaccurate but fluent outputs; on the other hand, our metrics are most apt at describing fluency, rather than correctness. This leads neural networks to "hallucinate", i.e., produce fluent but incorrect outputs that we currently struggle to detect automatically. For instance, Dopierre et al. (2021) report that when trying to produce a paraphrase for the input "*I am not sure where my phone is*", they obtain the following 'hallucination' behavior: "*How can I find the location of any Android mobile*". For many NLG applications, the correctness of an output is however mission critical. For instance, producing a plausible-sounding translation that is inconsistent with the source text puts in jeopardy the usefulness of a machine translation pipeline.

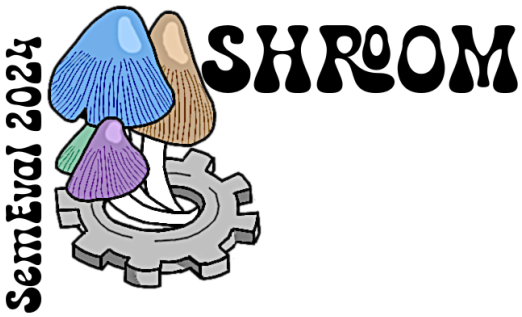


Figure 1: The SHROOM logo.

This motivates us to organize a **S**hared-task on **H**allucinations and **R**elated **O**bservable **O**vergeneration **M**istakes, or SHROOM. With our shared task, we hope to foster the growing interest in this topic in the community (e.g., Ji et al., 2023; Raunak et al., 2021; Guerreiro et al., 2023; Xiao and Wang, 2021; Guo et al., 2022). In particular, in the SHROOM we adopt a *post hoc* setting, where models have already been trained and outputs already produced. Participants were asked to perform binary classification to identify cases of **fluent overgeneration hallucinations** in two different setups: **model-aware** and **model-agnostic** tracks. That is, participants had to detect grammatically sound outputs which contain incorrect or unsupported semantic information, inconsistent

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1979–1993

with the reference input, with or without having access to the model that produced the output.

To that end, we constructed a dataset comprising a collection of checkpoints, inputs, references and outputs of systems covering three different NLG tasks: definition modeling (DM, Noraset et al., 2017), machine translation (MT) and paraphrase generation (PG) trained with varying degrees of accuracy. Datapoints were all annotated by 5 human annotators each resulting in 1000 validation items and 3000 test items.

Beyond simply detecting factually unsupported outputs, one of the goals of this shared task was to establish whether hallucinations are best construed as a categorical phenomenon or a gradient one. Similar remarks have been made with respect to textual entailment (Bowman et al., 2015). As such, participants' submission were scored both for accuracy (whether classifiers correctly identify hallucinations) and calibration (whether classifiers are confident about their prediction when they ought to be).

The shared task attracted a total of 58 different users grouped in 42 teams, out of which 26 elected to write a system description paper. Collectively, over the three weeks of the evaluation phase, participants submitted 300 valid sets of predictions on the model-aware track, and 320 on the model-agnostic track. We take this participation rate, along with the breadth of methodological approaches developed by participants, as clear signs of success for our shared task: This large pool of participants allows us to identify and discuss some key trends in how the task was tackled. Crucially, many participants rely on a handful of model, and often rely either on synthetic data for fine-tuning or zero-shot prompting strategies. In terms of raw performance, we note that while a majority of the teams (64 to 71%) did outperform our proposed baseline system, the performances of top-scoring systems are still consistent with a random handling of the more challenging items. In sum, this first iteration of the SHROOM underscores both an interest of the research community as well as the current limitations in our approaches.

The remainder of this article is structured as follows: Iin Section 2, we provide an overview of the current research landscape. Section 3 defines our theoretical framework, and Section 4 summarizes our data collection process. We then present and discuss shared task results in Sections 5 and 6 before concluding with a few thoughts on further research in Section 7.

2 Connecting with the past: related works and state of the art

It is now widely accepted that NLG models often generate outputs that are not faithful to the given input, commonly referred to in the community as hallucinations (Vinyals and Le, 2015; Raunak et al., 2021; Maynez et al., 2020). Yet there is minimal consensus on the optimal framework for its application. This lack of agreement is due in part to the diversity of tasks that NLG encompasses (Ji et al., 2023).

Guerreiro et al. (2023) propose a taxonomy of hallucinations that includes oscillatory productions, and fluent but strongly or fully "detached" outputs. While this taxonomy is well constructed, we find it inadequate for the needs of the community at large for four reasons: *(i)* It conflates some issues of fluency with semantic correctness (oscillatory productions are cases of non-fluent overgeneration where no extraneous semantic material is introduced); *(ii)* It only considers the most extreme cases of hallucinations (strongly or fully detached productions), whereas diagnosis of intermediary cases is bound to be more challenging and useful to the community; *(iii)* It focuses only on MT, although other tasks are also known to suffer from fluent overgeneration (e.g., Rohrbach et al., 2018), including the ones we propose to address; *(iv)* It uses only lowest scoring outputs, whereas any tool built to verify system outputs ought not to flag non-pathological outputs.

Alternative studies have built benchmarks for hallucination detection, with a predominant emphasis on dialogue systems. Li et al. (2023) propose the HaluEval benchmark using an annotation framework that does not necessarily center on the input given to the model and requires the annotators to search the internet for facts. Moreover, they opted to annotate the outputs of a popular LLM, with the major downsides that it is closed, not-transparent and commercial; rendering the research outputs that may stem from future studies less interesting. Other benchmarks include the works of Liu et al. (2022) and Zhou et al. (2021), which automatically insert hallucinations into training instances to generate syntactic data for token-level hallucination detection; Lin et al. (2022), which work with factual claims supported by reliable, publicly available

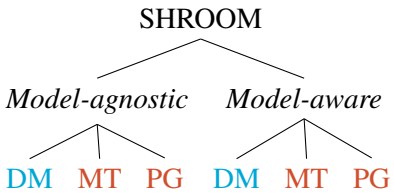


Figure 2: Shared task overview. Both tracks feature all three NLG tasks. Datapoints from systems in blue correspond to target-referential datapoints and in red the ones that are either target- or source-referential; which we refer to as *dual-referential*.

evidence; and Dziri et al. (2022), which focus on knowledge-based dialogue systems and base their annotation on NLI, relying only on the system's input, just as we do.

3 Tripping over hallucinations: task definition and annotation

In contrast with previous works (e.g. Guerreiro et al., 2023; Li et al., 2023), we focus on cases of fluent overgeneration since judgments pertaining to the over-generative nature of a production can be elicited by means of **inferential semantics**: if an output cannot be inferred from its semantic reference, then it contains some information that is not present in the reference—i.e., the model has generated more than we expected.[1] This approach connects with the theoretical framework sketched by van Deemter (2024), who likewise relies on inferential semantics but also considers undergeneration issues in NLG outputs. We provide multiple annotations and a gold majority label, given the low consensus on semantic annotations (Nie et al., 2020).

In Figure 2 we provide an overview of the task. The SHROOM is framed around two key distinctions: (i) model-aware vs. model-agnostic approaches, and (ii) source-referential vs. dual-referential datapoints. The former corresponds to whether participants have access to the model that generated the item: **Model-agnostic** approaches are practical, as models may not be accessible to end users; **Model-aware** approaches can lead to richer and more accurate diagnoses. The latter is a consequence of our inferential take on over-

[1]Note that if the output can be inferred from the reference but the information is not explicitly present in the reference, then the model is actually making a correct semantic inference: it is generating a semantically sound output. E.g., if the the model produces "my tie is blue" for the reference "my tie is the color of the sky", the model output is semantically sound.

```
{ "hyp":"A cigarette .",
  "ref":"tgt",
  "src":"I stepped outside to smoke myself a j .
         What is the meaning of J ?",
  "tgt":"( plural Js or J 's ) A marijuana
          cigarette .",
  "model":"ltg\/flan-t5-definition-en-base",
  "task":"DM",
  "labels":["Hallucination","Not Hallucination",
            "Not Hallucination", "Hallucination",
            "Hallucination"],
  "label":"Hallucination",
  "p(Hallucination)":0.6  }
```

Figure 3: Target-referential datapoint example from the validation set for the model-aware track.

generation: what can effectively serve as a semantic reference varies across NLP systems. For DM, where we fine-tune a language model to produce a definition for a given example of usage the datapoints are **target-referential**, i.e. the target is the sole usable semantic reference. In this context, the target serves as the sole usable semantic reference. Conversely, the target is expected to be semantically implied from the source in source-referential tasks, such as summarization. Note that we do not annotate source-referential tasks due to annotation challenges that make them unreliable for our purposes. In dual-referential tasks like PG & MT, this distinction bears no weight.

In Figure 3, we present an example datapoint displaying how we plan to encode all relevant information in a JSON format is provided. The datapoint keeps track of the source provided to the model as input (`src`), the intended target (`tgt`), the model production (`hyp`), the task this production was derived from (`task`), can correspond to DM, MT or PG), whether this datapoint is target-referential (`ref`), the annotations, the gold label and the proportion of annotators that labeled the utterance as a hallucination (`labels`, `label`, and `p(Hallucination)`). In the model-aware track, we will also provide a HuggingFace model name (`model`).

4 Foraging and harvesting season: Collected data

All SHROOM data (models, outputs and annotations) are available under a CC-BY license.[2]

[2]See helsinki-nlp.github.io/shroom

4.1 Data & model provenance

Participants have access to generated outputs from multiple systems trained to generate English output at various stages of their training, stemming from three sequence-to-sequence NLG tasks: DM, MT and PG. The SHROOM dataset consists of annotated *test* and *dev* sets, as well as a *unlabeled training split* of 30k datapoints per track and the full set of possible target references to allow corpus-wide approaches. To ensure effective annotation of the development and test sets, and to be able to guarantee a gradient in quality as measured by automated metrics, we pre-selected fluent outputs for the annotators, which we describe in the following.[3]

MT: For the model agnostic track we use the models from Mickus and Vázquez (2023). We compute perplexity for the all MT outputs and BERTScores with regards to the outputs and corresponding targets. We filter outputs with perplexity scores above the 2% quantile. From the filtered outputs, we randomly select 200 samples with BERTscores in the 1/7, 2/7, 3/7, 4/7, and 5/7 quantiles. For the model-aware track, we use the NLLB model (NLLB Team et al., 2022) and produce translations on the Flores-200 dataset from languages marked as low-resource to English. Next, we manually select a sample that is sufficiently fluent.

DM: We use the model of Segonne and Mickus (2023) for the model-agnostic track, and for the model-aware track we used the `flan-t5-definition-en-base` (Giulianelli et al., 2023). We generate outputs on the English portion of the CoDWoE dataset (Mickus et al., 2022), and manually select a sample that is reasonably fluent and contains no profanities.

PG: We used a pretrained and fine-tuned paraphrasing model[4] based on Pegasus (Zhang et al., 2020) for the model-aware track, and the controlled paraphrase generation model of Vahtola et al. (2023) for the model-agnostic track.

We generated paraphrase hypotheses using Europarl (Koehn, 2005) and Opusparcus (Creutz, 2018) for the model-aware and -agnostic tracks, respectively. For the model-aware setup, we generated 50 hypotheses for each source sentence using diverse beam search (Vijayakumar et al., 2016) using BLEU scores (Papineni et al., 2002) to select the least similar hypothesis for each source sentence to serve as its paraphrase. For the model-agnostic setup, we calculated control tokens for each source sentence as in Vahtola et al. (2023), scaled the length-controlling value in range (1, 1.5) with a uniform probability distribution to provoke hallucination in the generated sequences, and used beam search with a beam size of 5 to produce the paraphrases. We manually curated the final validation and test examples.

4.2 Annotation

We annotate a total of 4,000 items, which are split 25%–75% between development and test sets: 1000 datapoints come from PG, 1500 from DM and 1500 from MT. Each item is annotated by five annotators on whether the reference entails the output. Annotations are binary, for ease of dataset construction. Gold labels are defined with respect to the annotators' majority vote.

The annotators were enlisted via Prolific,[5] a paid platform specialized in gathering human data for research studies and AI dataset creation, among other purposes. We did not target any particular group of participants; the only screening prerequisites were that (i) participants had to be fluent in English and (ii) they should not have taken part in an initial pilot study.

We used Potato (Pei et al., 2022), an open-source annotation tool specifically designed to seamlessly integrate with Prolific. Annotators were first presented with a pre-annotation screen outlining the annotation guidelines, after which they commenced the annotation of items individually. Each item consisted of the Reference, the AI-generated output, and relevant context regarding the NLG task (DM, MT, or PG). The annotators were asked to answer the question *"Does the following AI output only contain information supported by the Reference?"* responding with either "yes" or "no," and were also given the opportunity to provide comments if necessary. Additionally, they could navigate back and forth through their assigned items. We set up a timer that notified the participants every 60 seconds of the time spent on an item. In Appendix B, we present a copy of the instructions we used.

To control for annotation quality, we manually reviewed annotations from two sets of selected an-

[3] Note that we do not warranty that the training split contains fluent outputs.

[4] https://huggingface.co/tuner007/pegasus_paraphrase

[5] https://www.prolific.com/

notators: (i) five randomly selected annotators; and (ii) the five annotators who completed the task the fastest (under 3.5 minutes). All 10 annotators completed 20 annotations each. We judged all 200 annotations to be sound, in that a reasoning could be reconstructed to explain the provided annotation.

Label distribution. Figure 4 provides an overview of the distribution of labels in the SHROOM dataset splits (validation and test), broken down per NLG task (MT, DM and PG) and track (model-aware vs. model-agnostic). In this figure, we consider the empirical probability that a given item is judged to be a hallucination, i.e., the proportion of annotators judging the NLG output is not supported by the intended semantic reference.

We can highlight two trends in this figure. The first one, and perhaps most important, is that hallucinations are not consensual among our annotators. If intuitions regarding hallucinations were clear-cut, we would strongly expect a bi-modal distribution of empirical label distributions being consistently judged as hallucinations or not hallucinations. Instead, we find a number of intermediate cases, where annotators are split: These account for 29–32% of the data, depending on the split (validation or test) and track (model-aware or model-agnostic). Given the small number of annotators per datapoint, we cannot confidently rule out the possibility of a sampling bias—it is plausible that a larger pool of annotator would yield more bimodal empirical distributions. On the other hand, this tentative evidence is also in line with what has been argued elsewhere for natural language inference (Nie et al., 2020; Zhou et al., 2022). This is in fact well exemplified by the datapoint provided in Figure 3: Whether the term *cigarette* is underspecified and can apply to any smokable substance, or whether it is to be understood as prototypically referring to tobacco cigarettes by default is, in fact, up for discussion—and it stands to reason that different speakers may form different opinions.

Second, it is difficult to find hallucinations: The higher the empirical probability, the fewer the datapoints. This is especially true in the PG task: these outputs rarely yields consensual hallucinations, whereas we can find such items in DM and MT much more frequently. Looking at the expected value of the empirical probability per task, we find that DM consistently ranks higher than MT, which in turns ranks higher than PG. Both of these differences are significant under a one-sided Mann-Whitney U-test in the two test tracks ($p < 0.0003$); in the model-aware validation dataset, only the difference between MT and PG is significant ($p < 2 \cdot 10^{-8}$), in the model-agnostic validation dataset, only the difference between DM and MT is ($p < 0.04$). We note that DM requires a more complex processing of its input, as it has to rely on facts captured by the underlying LLM during its pre-training phase; for MT and PG, the input of the NLG task contains the semantic information necessary to produce a valid output. As such, we conjecture that the difficulty of an NLG task fosters hallucinatory behavior.[6]

5 They got so high: shared task results

The competition was held via Codalab (Pavao et al., 2023). The leaderboard was left hidden during the evaluation phase (i.e., participants were not notified of their submissions' scores until the end of the evaluation phase) but users were allowed to make a high number of submissions (50).

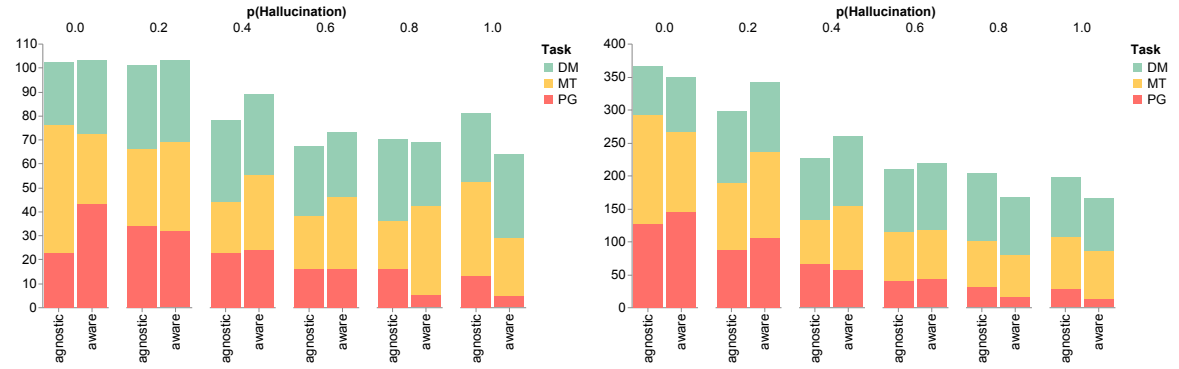
Systems are evaluated according to two criteria: the **accuracy** that the system reached on the binary classification, and their **calibration**, measured as the Spearman correlation of the systems' output probabilities with the proportion of the annotators marking the item as overgenerating. We rank systems by accuracy and break possible ties using calibration.

5.1 Baseline system

As a baseline for the task, we use an LLM[7] to evaluate whether the generated hypotheses are coherent with the provided context. Drawing upon Manakul et al. (2023), we use the prompt template listed in Figure 5. The system of Manakul et al. (2023), which has gathered some attention from the community, constitutes a straightforward approach based on a modern LLM, and is therefore well-suited to serve as a baseline in our shared-task:

[6] We also remark that the two tracks are broadly comparable in terms of hallucinatory content. Two-samples Kolmogorov-Smirnov tests for either split (test or validation) do not provide sufficient grounds to suggest a difference of distribution in labels between model-aware and model-agnostic tracks—which again suggests that the relevant difference is at the task level, rather than at the model level.

[7] We use quantized `Mistral-7B-Instruct-v0.2` (Jiang et al., 2023), from the `Hugging Face hub` `huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF` or the `llamacpp` project `github.com/ggerganov/llama.cpp`.



(a) Validation split

(b) Test split

Figure 4: Distributions of annotations

```
Context: {}
Sentence: {}
Is the Sentence supported by the Context above?
Answer using ONLY yes or no:
```

Figure 5: Prompt template used in the baseline system, adapted from Manakul et al. (2023).

it corresponds to a reasonable default approach to tackle the problem we challenge participants with.

The specific context varies depending on the task addressed, i.e. the source sentence for the paraphrase generation task, and the target sentence for machine translation and definition modeling tasks. As for the probability of hallucination, we rely on the probability assigned by the model to the first output word.[8] In cases where the output does not clearly indicate *yes* or *no*, we randomly select one, attributing a hallucination probability of 0.5.

On the model-agnostic track, our baseline system achieves an accuracy of 0.697 (with a calibration of $\rho = 0.403$), on the model-aware track, we observe an accuracy of 0.745 (with $\rho = 0.488$). We can also indicate some other simple heuristics, such as picking the most frequent label (viz., `Not Hallucination`): In this case, one would expect an accuracy of 0.593 on the model-agnostic track, and 0.633 on the model-aware track. A purely random guess between the two possible labels would result in an accuracy of 0.5. In short, our baseline systems systematically outperforms these crude heuristics.

5.2 Participating teams

A total of 59 individual users grouped in 42 teams participated in the shared task, out of which 26 elected to write a system description paper. During the evaluation phase, we received a total of 512 submissions, out of which 368 were successful. 264 of these submissions targeted both tracks, while 68 only targeted the model-agnostic track, and 36 only targeted the model-aware track. That is, we received 332 model-agnostic submissions and 300 model-aware submissions.

We present the model-agnostic track rankings in Table 1a and the model-aware track in Table 1b. As one might expect, there is a high correlation between the accuracy and calibration scores of each team's top ranking submission, which translates into a Spearman's ρ correlation of 0.909 on the model-agnostic track and 0.949 on the model-aware track. Most of the top submissions per team rank above our baseline ($30/42 \approx 71.4\%$ in the model-agnostic track, $25/39 \approx 64.1\%$ in the model-aware track). This appears roughly in line with all submissions globally: 69.9% of all model-agnostic submissions and 57.0% of all model-aware submissions score higher than our baseline.

Another point worth stressing is that teams that fare well on one track usually fare equally well on the other: For the 38 teams participating in both tracks, we find that the rank they obtain on the model-aware track correlates with the rank they obtain on the model-agnostic track (Spearman's $\rho = 0.884$). This would tentatively suggest that participants could not effectively leverage the supplementary data available in the model-aware track.[9]

Lastly, we note that there is a ceiling in terms

[8] We note that this simple heuristic may not accurately represent the true hallucination probability.

[9] An alternative account would be that all teams that participated in both tracks equally benefited from the access to the model weights, which we deem much less likely.

	team	Acc	ρ
1	Halu-NLP (Mehta et al., 2024)	0.847	0.770
2	OPDAI (Chen et al., 2024)	0.836	0.732
3	HIT-MI&T Lab (Liu et al., 2024)	0.831	0.768
4	SHROOM-INDElab (Allen et al., 2024)	0.829	0.721
5	Alejandro Mosquera	0.826	0.709
6	DeepPavlov (Belikova and Kosenko, 2024)	0.821	0.752
7	BruceW	0.821	0.735
8	TU Wien (Arzt et al., 2024)	0.817	0.737
9	SmurfCat (Rykov et al., 2024)	0.814	0.723
10	HaRMoNEE (Obiso et al., 2024)	0.814	0.626
11	AMEX AI LABS	0.813	0.728
12	Pollice Verso (Kobs et al., 2024)	0.803	0.676
13	MALTO (Borra et al., 2024)	0.801	0.681
14	UCC-NLP	0.795	0.664
15	Team CentreBack	0.792	0.623
16	Atresa	0.788	0.646
17	ustc_xsong	0.785	0.695
18	IRIT-Berger-Levrault (Bendahman et al., 2024)	0.783	0.636
19	silk_road	0.781	0.672
20	AILS-NTUA (Grigoriadou et al., 2024)	0.778	0.668
21	zhuming	0.773	0.481
22	SibNN	0.770	0.613
23	UMUTeam (Pan et al., 2024)	0.769	0.561
24	Noot Noot (Bahad et al., 2024)	0.765	0.584
25	HalluSafe (Rahimi et al., 2024)	0.763	0.629
26	Maha Bhaashya (Bhamidipati et al., 2024)	0.749	0.605
27	DUTh (Iordanidou et al., 2024)	0.744	0.475
28	Compos Mentis (Das and Srihari, 2024)	0.738	0.595
29	daixiang	0.737	0.583
30	NU-RU (Markchom et al., 2024)	0.728	0.595
	baseline system	0.697	0.403
31	SLPL SHROOM (Fallah et al., 2024)	0.694	0.423
32	Skoltech	0.684	0.674
33	CAISA	0.677	−0.430
34	AlphaIntellect (Choudhury et al., 2024)	0.654	0.295
35	deema	0.646	0.566
36	BrainLlama (Siino, 2024)	0.625	0.204
37	Byun (Byun, 2024)	0.617	0.239
38	Bolaca (Rösener et al., 2024)	0.613	0.217
	most frequent guess	0.593	
39	AI Blues	0.587	0.025
	random guess	0.500	
40	MARiA (Sanayei et al., 2024)	0.498	0.025
41	0x.Yuan	0.461	0.134

(a) Model-agnostic track rankings

	team	Acc	ρ
1	HaRMoNEE (Obiso et al., 2024)	0.813	0.699
2	Halu-NLP (Mehta et al., 2024)	0.806	0.715
3	TU Wien (Arzt et al., 2024)	0.806	0.707
4	OPDAI (Chen et al., 2024)	0.805	0.680
5	HIT-MI&T Lab (Liu et al., 2024)	0.805	0.712
6	SHROOM-INDElab (Allen et al., 2024)	0.802	0.656
7	AMEX AI LABS	0.801	0.696
8	DeepPavlov (Belikova and Kosenko, 2024)	0.799	0.713
9	silk_road	0.798	0.687
10	AILS-NTUA (Grigoriadou et al., 2024)	0.795	0.685
11	BruceW	0.794	0.660
12	Team CentreBack	0.789	0.606
13	UCC-NLP	0.789	0.644
14	ustc_xsong	0.787	0.658
15	UMUTeam (Pan et al., 2024)	0.784	0.507
16	HalluSafe (Rahimi et al., 2024)	0.783	0.537
17	SmurfCat (Rykov et al., 2024)	0.783	0.671
18	Atresa	0.783	0.624
19	IRIT-Berger-Levrault (Bendahman et al., 2024)	0.781	0.601
20	Pollice Verso (Kobs et al., 2024)	0.777	0.601
21	NU-RU (Markchom et al., 2024)	0.768	0.582
22	zhuming	0.768	0.472
23	SibNN	0.763	0.587
24	Compos Mentis (Das and Srihari, 2024)	0.756	0.566
25	DUTh (Iordanidou et al., 2024)	0.755	0.528
	baseline system	0.745	0.488
26	AlphaIntellect (Choudhury et al., 2024)	0.711	0.426
27	SLPL SHROOM (Fallah et al., 2024)	0.706	0.426
28	deema	0.688	0.519
29	BrainLlama (Siino, 2024)	0.671	0.244
30	daixiang	0.649	0.218
	most frequent guess	0.633	
31	Bolaca (Rösener et al., 2024)	0.626	0.283
32	Noot Noot (Bahad et al., 2024)	0.613	0.355
33	Byun (Byun, 2024)	0.610	0.234
34	Maha Bhaashya (Bhamidipati et al., 2024)	0.606	0.209
35	CAISA	0.567	−0.100
36	Skoltech	0.557	−0.011
37	MARiA (Sanayei et al., 2024)	0.505	0.009
	random guess	0.500	
38	octavianB (Brodoceanu, 2024)	0.483	−0.064

(b) Model-aware track rankings

Table 1: SHROOM team rankings. Codalab usernames are used to define teams when no other information was provided.

of performances: The most effective systems misclassify between 15 to 19% of all items, or almost one in every six or five datapoints. We have discussed above that, as hallucinations are a graded phenomenon, a large segment of our data (30%) corresponds to ambiguous cases where annotators are split 2 vs. 3. As such, it is worth stressing that top scores are consistent with models that classify consensual items well (where at most one annotator disagree), but perform at random chance on the more challenging ambiguous datapoints.

6 A bunch of fun guys: qualitative analysis of participants systems

We derive our analyses from system description papers as well as self-reports from a handful of participants who elected to not provide a full description of their systems. This corresponds to 33 systems

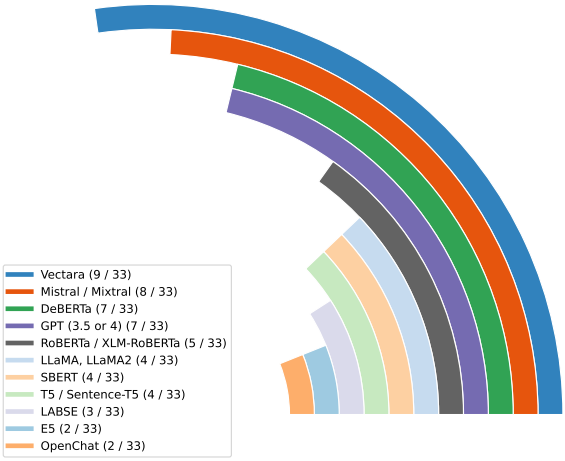


Figure 6: Known models used by more than one team. A full circle would correspond to a given model used by all of respondents, half a circle to 50% of respondents using said model. Best viewed in color.

out of the 42 identified teams that participated to the shared task, out of which 7 did not provide a full description. See also Table 2 in Appendix A for further details.

How the task was approached. The teams used a variety of methods to address the problem, ranging from ensemble techniques to fine-tuning pre-trained language models (LLMs) and prompt engineering. As expected, most teams used popular pre-trained LLMs such as GPT, LLaMA, DeBERTa, RoBERTa, and XLM-RoBERTa; Figure 6 provides a summary of which models were most popular among our teams. The Vectara hallucination evaluation model[10] turned out to be extremely popular, as more than 1 in 4 teams that provided information about their systems report having used it in their experiments. If we add other DeBERTa-based models, this number climbs to $16/33$, i.e. almost every other team used DeBERTa or a variant thereof.

Yet, the ways in which these LLMs were used cover a wide range of approaches: Some either fine-tuned on hallucination data or optimized with prompts; others employed in-context learning with role-playing, automatic prompt generation, and ensemble methods. Furthermore, some teams focused on zero-shot and few-shot approaches, while others focused on synthetic data generation and semi-supervised learning techniques to construct a labeled training set. Especially noteworthy, Rahimi et al. (2024) report constructing a manual dataset of 3000 datapoints for training their systems.

Teams predominantly relied on the data constructed for the SHROOM, although some teams added datasets such as QQP and PAWS. Interestingly, we also note five teams relying on NLI/entailment data or models, including some that achieved high results (Obiso et al., 2024; Sanayei et al., 2024; Borra et al., 2024; Liu et al., 2024 and Team Centre-Back)—and this matches the theoretical framework adopted in this shared task.

What worked well. We now turn to what distinguishes top scorers from other submissions. We note that systems based on the closed-source models GPT-3.5 and GPT-4 tend to fare well: 4 out of the 6 highest scoring systems on either track—Mehta et al. (2024); Obiso et al. (2024); Liu et al. (2024); Allen et al. (2024) and Alejandro Mosquera—all report using these models. This is however not a strict requisite as OPDAI (Chen et al., 2024) manages to rank high (2^{nd} on the model-agnostic track and 4^{th} on the model-aware track) without it. Neither does using closed-source models guarantee a high result: UCC-NLP and Markchom et al. (2024) also use GPT-3.5, and while the former is ranked 14^{th} on the model-agnostic track and 13^{th} on the model-aware track, the latter is ranked 30^{th} on the model-agnostic track and 21^{st} on the model-aware track, and only outperforms the baseline model in accuracy by 0.02 to 0.03 points.

Remarkably, many of the top-scoring approaches rely on fine-tuning (Liu et al., 2024; Obiso et al., 2024; Arzt et al., 2024; Chen et al., 2024) or ensembling (Mehta et al., 2024; Belikova and Kosenko, 2024, Alejandro Mosquera), suggesting that high performances do not come out of the box from off-the-shelf LLMs and systems. It is necessary to adapt existing models or establish to what extent their predictions are useful to the task at hand.

Another important trend we identify is that the number of submissions per team anti-correlates with the rank they obtain: The more participants submitted, the higher their best scores went. This is visualized in Figure 7: On both tracks, we find reasonable anti-correlations ($-0.58 < \rho < -0.44$) indicating that top-scorers tended to submit more. This might provide an alternative explanation for what distinguishes top-scorers from other participants: If we were to model participants' submissions as a random process, we would expect that sampling more often (i.e., submitting more) would mechanically yield a better rank.

[10] https://huggingface.co/vectara/hallucination_evaluation_model

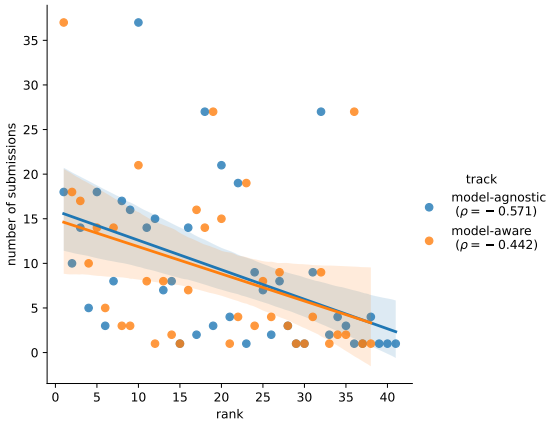


Figure 7: Rank obtained vs. number of submissions made on both tracks.

Overall, the high methodological diversity highlights the complexity of hallucination detection, even when contained the simple inferential semantics framework of our shared task: While a focus on NLI or using high-performance closed source models may help, the highest scores are obtained through thorough involvement—both in terms of model training and prediction set submissions.

7 Much room to grow: conclusions and future perspectives

This first iteration of the SHROOM shared task on detecting hallucinations has allowed us to make significant headway into understanding the confabulatory behavior of modern NLG systems. The data collected demonstrate that *hallucinations correspond to a gradient phenomenon*, and that different speakers form different opinions as to what counts as a hallucination. We were also able to showcase that *ambiguous items remain challenging*, and that the current state of the art on the dataset we provided is compatible with simple random guesses whenever the data is more ambiguous. This results underscore the massive gap that NLP research urgently needs to address: one out of every six items is still misclassified by the most effective systems showcased during this shared task.

The diversity of methodologies employed by participants underscores how *out-of-the-box solutions are not sufficient*: Highest scoring teams had to rely on fine-tuning or ensembling and made a high number of submissions. Relatedly, *access to the model parameters was of limited help*: Few approaches attempted to perform model-specific investigations, and performances on the model-aware track are in fact lower than what we observed on the model-agnostic track. Properly leveraging the parameter space for finer-grained hallucination detection remains a point for future research to investigate.

This shared task has not broached some crucial aspects and questions: How do these results translate insofar as modern LLMs—often much larger and better trained than the systems we studied here—are concerned? Can we leverage sentence-level predictions to pinpoint token-level issues with the output of our NLG systems? And will the difficulties that we underscored in this purely English be exacerbated when studying other languages—especially those that are less well-resourced and typologically different? Answering these questions and more will require further research—and perhaps future iterations of this shared task.

Overall, the success of this shared task is owed to its committed participants. We received over 350 submissions in the span of three weeks from across the world. The width of approaches studied and reported upon provides a useful snapshot of where the field is at, what approaches are favored, and what gaps still need to be overcome. We expect that the results of the SHROOM will provide a useful starting point for future work on hallucinations.

Doing SHROOM responsibly: ethical considerations

We strive to adhere to the ACL Code of Ethics.

Broader Impact. Hallucinated outputs from large language models can be used to further spread disinformation and advance misleading narratives. Detecting hallucinated outputs is an important step in elucidating the factors of this phenomena and contribute to ongoing efforts to mitigate hallucination. This leads to the development of more trustworthy generative language models.

Data and Annotators. Our annotators were suitably compensated for their work in excess of minimum wage. Due to the nature of the proposed task, the data we release might contain false or misleading statements. In the case of annotated data, these statements are labeled as such, but this does not for the unannotated portions of the data. We manually pre-filtered the data to remove profanities before providing them to annotators. Such precautions were not taken for the unannotated portion of the dataset, which might therefore contain offensive, obscene or otherwise unconscionable items.

Acknowledgments

The construction of the SHROOM dataset was made possible by a grant from the Oskar Öfflund Foundation. This work is also supported by the ICT 2023 project "Uncertainty-aware neural language models" funded by the Academy of Finland (grant agreement № 345999). We also thank the CSC-IT Center for Science Ltd., for computational resources.

The shared task logo (cf. Figure 1) uses the "Retro Cool" font from Nirmana Visual (https://nirmanavisual.com/), made available for personal / non-commercial uses.

References

Bradley Allen, Fina Polat, and Paul Groth. 2024. Shroom-indelab at semeval-2024 task 6: Zero- and few-shot llm-based classification for hallucination detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 826–831, Mexico City, Mexico. Association for Computational Linguistics.

Varvara Arzt, Mohammad Mahdi Azarbeik, Ilya Lasy, Tilman Kerl, and Gábor Recski. 2024. Tu wien at semeval-2024 task 6: Unifying model-agnostic and model-aware techniques for hallucination detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1172–1186, Mexico City, Mexico. Association for Computational Linguistics.

Sankalp Sanjay Bahad, Yash Bhaskar, and Parameswari Krishnamurthy. 2024. Nootnoot at semeval-2024 task 6: Hallucinations and related observable overgeneration mistakes detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 964–968, Mexico City, Mexico. Association for Computational Linguistics.

Julia Belikova and Dmitrii Kosenko. 2024. Deeppavlov at semeval-2024 task 3: Multimodal large language models in emotion reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1757–1767, Mexico City, Mexico. Association for Computational Linguistics.

Nihed Bendahman, Karen Pinel-Sauvagnat, Gilles Hubert, and Mokhtar BILLAMI. 2024. Irit-berger-levrault at semeval-2024: How sensitive sentence embeddings are to hallucinations? In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 560–565, Mexico City, Mexico. Association for Computational Linguistics.

Patanjali Bhamidipati, Advaith Malladi, Manish Shrivastava, and Radhika Mamidi. 2024. Maha bhaashya at semeval-2024 task 6: Zero-shot multi-task hallucination detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1709–1713, Mexico City, Mexico. Association for Computational Linguistics.

Federico Borra, Claudio Savelli, Giacomo Rosso, Alkis Koudounas, and Flavio Giobergia. 2024. Malto at semeval-2024 task 6: Leveraging synthetic data for llm hallucination detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1688–1694, Mexico City, Mexico. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Octavian Brodoceanu. 2024. octavianb at semeval-2024 task 6: An exploration of humanlike qualities of hallucinated llm texts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1149–1154, Mexico City, Mexico. Association for Computational Linguistics.

Cheolyeon Byun. 2024. Semeval2024 task6 group byun. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 269–272, Mexico City, Mexico. Association for Computational Linguistics.

Ze Chen, Chengcheng Wei, Songtan Fang, Jiarong He, and Max Gao. 2024. Opdai at semeval-2024 task 6: Small llms can accelerate hallucination detection with weakly supervised data. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 707–715, Mexico City, Mexico. Association for Computational Linguistics.

Sohan Choudhury, Priyam Saha, Subharthi Ray, Shankha Shubhra Das, and Dipankar Das. 2024. Alphaintellect at semeval-2024 task 6: Detection of hallucinations in generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 939–945, Mexico City, Mexico. Association for Computational Linguistics.

Mathias Creutz. 2018. Open subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Souvik Das and Rohini Srihari. 2024. Compos mentis at semeval2024 task6: A multi-faceted role-based large language model ensemble to detect hallucination. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1459–1464, Mexico City, Mexico. Association for Computational Linguistics.

Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. PROTAUGMENT: Unsupervised diverse short-texts paraphrasing for intent detection meta-learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2454–2466, Online. Association for Computational Linguistics.

Nouha Dziri, Hannah Rashkin, Tal Linzen, and David Reitter. 2022. Evaluating attribution in dialogue systems: The BEGIN benchmark. *Transactions of the Association for Computational Linguistics*, 10:1066–1083.

Pouya Fallah, Soroush Gooran, Mohammad Jafarinasab, Pouya Sadeghi, Reza Farnia, Amirreza Tarabkhah, Zeinab Sadat Taghavi, and Hossein Sameti. 2024. Slpl shroom at semeval2024 task 06 : A comprehensive study on models ability to detect hallucination. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1137–1143, Mexico City, Mexico. Association for Computational Linguistics.

Mario Giulianelli, Iris Luden, Raquel Fernandez, and Andrey Kutuzov. 2023. Interpretable word sense representations via definition generation: The case of semantic change analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3130–3148, Toronto, Canada. Association for Computational Linguistics.

Natalia Grigoriadou, Maria Lymperaiou, George Filandrianos, and Giorgos Stamou. 2024. Ails-ntua at semeval-2024 task 6: Efficient model tuning for hallucination detection and analysis. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1559–1570, Mexico City, Mexico. Association for Computational Linguistics.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia. Association for Computational Linguistics.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Ioanna Iordanidou, Ioannis Maslaris, and Avi Arampatzis. 2024. Duth at semeval-2024 task 6: Comparing pre-trained models on sentence similarity evaluation for detecting of hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1053–1059, Mexico City, Mexico. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Konstantin Kobs, Jan Pfister, and Andreas Hotho. 2024. Pollice verso at semeval-2024 task 6: The roman empire strikes back. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1539–1546, Mexico City, Mexico. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Wei Liu, Wanyao Shi, Zijian Zhang, and Hui Huang. 2024. Hit-mi&t lab at semeval-2024 task 6: Deberta-based entailment model is a reliable hallucination detector. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1798–1808, Mexico City, Mexico. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Thanet Markchom, Subin Jung, and Huizhi Liang. 2024. Nu-ru at semeval-2024 task 6: Hallucination and related observable overgeneration mistake detection using hypothesis-target similarity and self-checkgpt. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 253–260, Mexico City, Mexico. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Rahul Mehta, Andrew Hoblitzell, Jack O'Keefe, Hyeju Jang, and Vasudeva Varma. 2024. Halu-nlp at semeval-2024 task 6: Metacheckgpt - a multi-task hallucination detection using llm uncertainty and meta-models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 335–341, Mexico City, Mexico. Association for Computational Linguistics.

Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1–14, Seattle, United States. Association for Computational Linguistics.

Timothee Mickus and Raúl Vázquez. 2023. Why bother with geometry? on the relevance of linear decompositions of transformer embeddings. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 127–141, Singapore. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3259–3266. AAAI Press.

Timothy Obiso, Jingxuan Tu, and James Pustejovsky. 2024. Harmonee at semeval-2024 task 6: Tuning-based approaches to hallucination recognition. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1311–1320, Mexico City, Mexico. Association for Computational Linguistics.

Ronghao Pan, José Antonio García-Díaz, Tomás Bernal-Beltrán, and Rafael Valencia-García. 2024. Umuteam at semeval-2024 task 6: Leveraging zero-shot learning for detecting hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 661–667, Mexico City, Mexico. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2023. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6.

Jiaxin Pei, Aparna Ananthasubramaniam, Xingyao Wang, Naitian Zhou, Apostolos Dedeloudis, Jackson Sargent, and David Jurgens. 2022. POTATO: The portable text annotation tool. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 327–337, Abu Dhabi, UAE. Association for Computational Linguistics.

Zahra Rahimi, Hamidreza Amirzadeh, Alireza Sohrabi, Zeinab Taghavi, and Hossein Sameti. 2024. Hallusafe at semeval-2024 task 6: An nli-based approach to make llms safer by better detecting hallucinations and overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 139–147, Mexico City, Mexico. Association for Computational Linguistics.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

Elisei Sergeevich Rykov, Yana Shishkina, Ksenia Petrushina, Ksenia Titova, Sergey Petrakov, and Alexander Panchenko. 2024. Smurfcat at semeval-2024 task 6: Leveraging synthetic data for hallucination detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 869–880, Mexico City, Mexico. Association for Computational Linguistics.

Béla Linus Rösener, Hong-Bo Wei, and Ilinca Vandici. 2024. Team bolaca at semeval-2024 task 6: Sentence-transformers are all you need. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1687–1690, Mexico City, Mexico. Association for Computational Linguistics.

Reza Sanayei, Abhyuday Singh, MohammadHossein Rezaei, and Steven Bethard. 2024. Maria at semeval 2024 task-6: Hallucination detection through llms, mnli, and cosine similarity. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1594–1598, Mexico City, Mexico. Association for Computational Linguistics.

Vincent Segonne and Timothee Mickus. 2023. Definition modeling : To model definitions. generating definitions with little to no semantics. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 258–266, Nancy, France. Association for Computational Linguistics.

Marco Siino. 2024. Brainllama at semeval-2024 task 6: Prompting llama to detect hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 82–87, Mexico City, Mexico. Association for Computational Linguistics.

Teemu Vahtola, Mathias Creutz, and Jrg Tiedemann. 2023. Guiding zero-shot paraphrase generation with fine-grained control tokens. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 323–337, Toronto, Canada. Association for Computational Linguistics.

Kees van Deemter. 2024. The Pitfalls of Defining Hallucination. *Computational Linguistics*, pages 1–10.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

Xiang Zhou, Yixin Nie, and Mohit Bansal. 2022. Distributed NLI: Learning to predict human opinion distributions for language reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 972–987, Dublin, Ireland. Association for Computational Linguistics.

A Shared consciousnesses: Overview of approaches used by SHROOM teams

In Table 2, we provide a short overview of the various teams, the resources they utilized (models & datasets), as well as a short description of their approach.

B What SHROOM makes you do: Annotation guidelines

In Figure 8, we provide an exact copy of the annotation guidelines given to the annotators. These guidelines are based on five of the organizers' experience of annotating the trial set, and were provided to annotators recruited for the validation and test splits.

Annotation guidelines for SHROOM

Thank you for agreeing to participate in the SHROOM survey! The present document will provide you some general guidelines as to what is expected of you.

What are we looking for?

Our goal is to assess the **truthfulness** of sentences and documents written with artificial intelligence. In particular, we are interested in cases when the **AI outputs** are not supported by the facts. Such unsupported outputs are colloquially referred to as **'hallucinations'**.
You will be presented with a series of items. Each item contains an **AI output**, as well as one or more **References**, i.e., examples of what the model should have produced in an ideal scenario. Outputs and references can correspond to different types of sentences and paragraphs: news headlines, dictionary definitions, movie subtitles…
We ask that **you mark for every item**, whether the output contains or describes facts that are **not supported by the provided reference**. Such items are instances of hallucinations.

In other words: **hallucinations are cases where the AI output is more specific than it should be, given the available reference**.

Some Examples and counterexamples Hallucinations

Hallucinations

Below are some examples of items our research team unanimously considered as hallucinations:

Example 1

```
Reference: The worship of trees.

AI output: (uncountable) The study of trees.
```

Example 2

```
    Reference: Why is everyone laughing?
    AI output: Why is everyone okay?
```

Example 3

```
    Reference: You’re a scam artist.
    AI output: You’re not a good scam artist.
```

Not Hallucinations

And next, here are a few items that we unanimously did not consider as hallucinations:

Example 1

```
    Reference: Capable of being deployed by parachute.
    AI output: Capable of being parachuted.
```

Example 2

```
    Reference: When did you see him?
    AI output:When was the last time you saw him?
```

One last thing: Hallucinations vs. undergeneration

In some instances, the AI output can contain **less** information than the reference. We refer to such items as cases of **“undergeneration”**. These are not necessarily cases of hallucinations: as long as what is stated in the AI output is supported by the reference, such items should not be marked as hallucinations.

Here is one straightforward example:

```
Reference: I can't do it alone. You have to help me.
AI output:I can't do it alone.
```

Given that all the information present in the AI output is also found in the reference, **this should not be marked as a hallucination**.

That's it!

Figure 8: Annotation guidelines.

Team & Paper	Resources	Overview
AI Blues		*(No report)*
AILS-NTUA Grigoriadou et al. (2024)	SHROOM datasets; Vectara model.	Fine-tuned models and voting classifier.
Alejandro Mosquera	SHROOM datasets; COMET, Vectara, LaBSE, GPT35 and GPT4 models.	Ensemble of publicly available models. Logistic Regression was used as final scoring model.
AlphaIntellect Choudhury et al. (2024)	SHROOM dataset, SBERT	Fully-connected neural network classifiers with SBERT embeddings as input.
AMEX AI LABS	SHROOM datasets; Vectara and OpenChat models.	Ensemble of LLM (using Openchat) zero shot and few shot with Vectara cross encoder based scores.
Atresa		*(No report)*
Bolaca Rösener et al. (2024)	SHROOM dataset, SBERT	Logistic regression and feed-forward classifier trained on SBERT embeddings
BrainLlama Siino (2024)	LLaMA model.	Prompt-based approach with LLaMA.
BruceW		*(No report)*
Byun Byun (2024)	SHROOM dataset, data augmentation, RoBERTa	Finetuned a BERT or RoBERTa model with a softmax layer to output the probability of hallucinated text. Finetuning data is the labelled SHROOM data augmented with data points constructed by replacing words with synonyms.
CAISA		*(No report)*
Compos Mentis Das and Srihari (2024)	HalluEval dataset; Mistral 7B instruct model.	Ensemble of several role-based LLMs, which were either fine-tuned on hallucination data or role-based prompting.
daixiang		*(No report)*
deema		*(No report)*
DeepPavlov Belikova and Kosenko (2024)	SHROOM dataset; OpenChat, DeBERTa, RoBERTa and T5 models.	Ensemble of several pretrained Transformer-based models to get features for validation and test data of SHROOM dataset and trained a boosting-based meta-model on top.
DUTh Iordanidou et al. (2024)	SHROOM, LaBSE, T5, DistilUSE	Using pre-trained LLMs and classifiers
HalluSafe Rahimi et al. (2024)	SHROOM, labeled 3000 samples of the training data	Fine-tuned a DeBERTa-v3-large
Halu-NLP Mehta et al. (2024)	SHROOM datasets; GPT, SelfCheckGPT and Vectara models.	Prompts and GroupCheckGPT. NB: due to a team name change, this team is also referred to as GroupCheckGPT by some participants.
HaRMoNEE Obiso et al. (2024)	SHROOM, SNLI, MNLI and PAWS datasets; Vectara and GPT4 models.	Highest results obtained with zero-shot prompting in the model-aware track; pretraining on NLI and PAWS followed by finetuning on the model-agnostic track.
HIT-MI&T Lab Liu et al. (2024)	SHROOM with training dataset labeled using GPT-4; DeBERTaV3, InternLM2, SBERT, and UniEval.	Fine-tune the DeBERTaV3 and InternLM2 models, and call the SBERT and UniEval models to select the optimal threshold usinf SHROOM & syntheticaly labeled data. The system obtians the final results by combining the prediction results of each model.
IRIT-Berger-Levrault Bendahman et al. (2024)	SHROOM datasets; Sentence-t5, BGE, e5 models.	Computes the cosine similarity of sentence embeddings and classify based on an empirical threshold value.
Maha Bhaashya Bhamidipati et al. (2024)	DeBERTa models.	Zero shot inference, pretrained cross encoder model
MALTO Borra et al. (2024)	SHROOM model-agnostic dataset, DeBERTa pretrained and finetuned on MNLI, SOLAR-10.7B quantized from TheBloke (for synthetic data generation)	Encoder and classifier, fine-tuned in various ways (including with synthetic data)
MARiA Sanayei et al. (2024)	SHROOM dataset, SBERT, bart-large-mnli, Mixtral	Three approaches: (1) Cosine similarity of SBERT embeddings between source-hypothesis and source-target pairs; (2) NLI classification using bart-large-mnli model; and (3) Mixtral prompting. Only the Mixtral results were submitted.
Noot Noot Bahad et al. (2024)	SHROOM dataset; Mixtral and RoBERTa models.	Mixtral prompting and RoBERTa finetuning.
NU-RU Markchom et al. (2024)	SHROOM, GPT-3.5, Sentence Transformers	Tried two approaches: (1) hypothesis-target cosine similarity, using a threshold value to determine whether the hypothesis is a hallucation. (2) SelfCheckGPT with a customized prompt for each NLG task, designed to assess its coherence with the provided source and target. Each prompt is iterated through the GPT-3.5 model five times, and the final label is determined by the majority response.
octavianB Brodoceanu (2024)	RoBERTa	Used a pretrained model (roberta-large-openai-detector) that has been trained to distinguish between text generated by LLMs and text written by humans.
OPDAI Chen et al. (2024)	SHROOM, Mistral-7B-Instruct-v0.2, self constructed training data	Supervised fine-tuning over synthetically constructed weakly supervised training data.
Pollice Verso Kobs et al. (2024)	Mistral2, LLaMa2, Phi2 and Zephyr models; uses SHROOM train set for prompt optimization.	Ensembling over the output logits of prompt-based LLMs (mistral, llama etc) after automatically optimizing their prompts ("OPRO").
SHROOM-INDElab Allen et al. (2024)	SHROOM dataset; GPT 3.5 and GPT 4 models.	In-context learning with role-play and automatic prompt generation in a few-shot classfier, using a closed-source LLM.
SibNN	SHROOM datasets; XLM-RoBERTa model.	Fine-tunes a self-adaptive hierarchical variant of XLM-RoBERTa-XL twice: first as an embedder (in a few-shot mode), then as a binary classifier. More details at https://huggingface.co/bond005/xlm-roberta-xl-hallucination-detector.
silk_road	SHROOM datasets; Vectara model.	Fine-tunes an off-the-shelf Cross-Encoder hallucination evaluation model.
Skoltech		*(No report)*
SLPL SHROOM Fallah et al. (2024)	SHROOM datasets; LaBSE, DeBERTa, Zephyr, Mistral and Llama2 models.	Using two LLMs to classify and explain their decision and another LLM to judge and decide based on those explanations.
SmurfCat Rykov et al. (2024)	SHROOM (synthetically augmented), QQP and PAWS datasets; E5, T5, Vectara models.	Fine-tuning of e5-mistral-7b-instruct using synthetic data collected with LLaMA2-7B adapters trained to produce data with and without hallucinations. However, there are two other systems: one works as a voting ensemble of multiple LLMs, and another uses the Mutual Implication Score architecture.
Team CentreBack	SHROOM dataset; DeBERTa model.	Uses an off-the-shelf library (SelfCheckGPT's SelfCheckNLI function) to calculate contradiction scores on a small labeled test set and then defined a threshold for hallucination.
TU Wien Arzt et al. (2024)	SHROOM dataset; Vectara model.	Model-aware track best submissions uses a Vectara hallucination detection model finetuned on the validation set. The best model-agnostic track submission is a meta-model that utilizes linear regression and is trained on features that correspond to probabilities predicted by individual systems we implemented.
UCC-NLP	SHROOM dataset; GPT-3.5 and Vectara models.	Uses BertScore and GPT-3.5 to create synthetic labels and fine-tune a Vectara LLM.
UMUTeam Pan et al. (2024)	SHROOM dataset; TULU-DPO model.	Zero-shot approach
ustc_xsong		*(No report)*
zhuming		*(No report)*
0x.Yuan	Mistral, Mixtral, LLaMA, Falcon, WizardLM and Capybara models.	Zero-shot prompt engineering. Expects most LLMs will have different hallucination patterns, and tests whether ensembling can mitigate this.

Table 2: Participating teams and their respective works.

SemEval-2024 Task 9: BRAINTEASER: A Novel Task Defying Common Sense

Yifan Jiang[1], **Filip Ilievski**[1,2], **Kaixin Ma**[3]
[1] Information Sciences Institute, Viterbi School of Engineering, University of Southern California
[2]Department of Computer Science, Faculty of Science, Vrije Universiteit Amsterdam
[3]Tencent AI Lab, Bellevue, WA
yifjia@isi.edu, f.ilievski@vu.nl, kaixinma@global.tencent.com

Abstract

While vertical thinking relies on logical and commonsense reasoning, lateral thinking requires systems to defy commonsense associations and overwrite them through unconventional thinking. Lateral thinking has been shown to be challenging for current models but has received little attention. A recent benchmark, BRAINTEASER, aims to evaluate current models' lateral thinking ability in a zero-shot setting. In this paper, we split the original benchmark to also support fine-tuning setting and present SemEval Task 9: **BRAINTEASER(S)**,[1] the first task at this competition designed to test the system's reasoning and lateral thinking ability. As a popular task, BRAINTEASER(S)'s two subtasks receive 483 team submissions from 182 participants during the competition. This paper provides a fine-grained system analysis of the competition results, together with a reflection on what this means for the ability of the systems to reason laterally. We hope that the BRAINTEASER(S) subtasks and findings in this paper can stimulate future work on lateral thinking and robust reasoning by computational models.

1 Introduction

Vertical thinking requires logical and commonsense reasoning, i.e., making plausible sequential associations of different pieces of commonsense knowledge. As presented in Figure 1 (top), we can easily infer that flooding a room requires filling it with water, based on common sense, and inanimate objects with five fingers are gloves in the riddle. In contrast, lateral thinking is a creative and divergent process that requires thinking out of the box and defying common sense. For example, as shown in Figure 1 (bottom), one needs to overwrite the commonsense associations of *man shaves* to *he shaves himself*, and regard the man as somebody who shaves others all day (e.g., a barber) to answer the lateral puzzle.

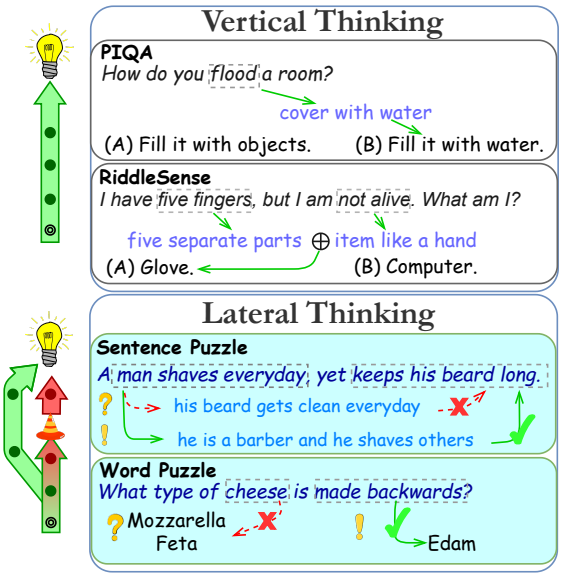


Figure 1: Figure from the first lateral thinking benchmark BRAINTEASER (Jiang et al., 2023c), contrasting existing Vertical Thinking tasks (PIQA (Bisk et al., 2020) and RiddleSense (Lin et al., 2021)) to lateral thinking. Solving BRAINTEASER's lateral puzzles requires default commonsense thinking to be deprecated.

While there are many datasets focusing on commonsense reasoning (Talmor et al., 2019; Bisk et al., 2020; Sap et al., 2019b) and numerous studies on improving commonsense reasoning ability of artificial systems (Ma et al., 2021a,b; Zhang et al., 2022), lateral thinking challenges have received little attention and are often filtered out as noise during preprocessing (Vajjala and Meurers, 2012; Speer et al., 2017; Sap et al., 2019a). Consequently, artificial systems' ability to solve lateral thinking problems remains understudied.

To bridge this gap, in (Jiang et al., 2023c), we introduce a novel BRAINTEASER benchmark with two tasks of different granularity: Sentence Puz-

[1]We use BRAINTEASER to represent the original benchmark and BRAINTEASER(S) to represent the data in SemEval task for clarity.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1994–2008

zles and Word Puzzles (cf. Figure 1). The task is formulated in a multiple-choice QA setting for a straightforward human and automatic evaluation. The dataset is constructed via a three-stage pipeline to ensure that the questions are valid and challenging.

We organize our SemEval Task with **BRAINTEASER(S)**, which contains the same data as the BRAINTEASER benchmark to *study model's lateral thinking ability*. Differing from the original benchmark that only focuses on the zero-shot setting, BRAINTEASER(S) divides this data into train/trial/test sets and has no limitation on the method adaptation. The goal of this paper is to describe the SemEval task and provide an analysis of the participant results. We provide details of the data construction pipeline in Section 2 and the SemEval Task description in Section 3. We present the overall leaderboard result and fine-grained method analysis in Section 4. Finally, we discuss the summarized result and conclude with high-level insight to stimulate future works on lateral thinking. For further information, we refer the reader to our source code,[2] task website,[3] and competition website.[4]

2 Source Dataset

We use our recently introduced BRAINTEASER dataset (Jiang et al., 2023c) as the basis for our evaluation. In this section, we briefly describe the data construction pipeline and we refer interested readers to (Jiang et al., 2023c) for full details.

The data construction pipeline has three stages. In the first stage, we collect lateral thinking puzzles from public websites such as `riddles.com` and `rd.com` and conduct filtering and deduplication. Then, the remaining questions are manually verified to ensure that they fit in the sentence or word puzzle categories.

Since the collected puzzles are open-ended questions, which poses great challenges for evaluation. These open-ended puzzles are then converted to multiple-choice questions in the second stage. Specifically, we leverage tools such as COMET (Hwang et al., 2021), WordNet and Wikipedia to construct distractors for every question. For sentence puzzles, we collect distractors that overwrite non-central premises of the question, and for word puzzles, we collect distractors that are semantically similar to the correct answer to ensure they are challenging for systems.

Table 1: Key statistics of the BRAINTEASER dataset. Choices combine the correct answer with all the distractors.

	Sentence	**Word**
# Puzzles	627	492
Average Question Tokens	34.88	10.65
% Long Question (>30 tokens)	48.32%	2.23%
Average Answer Tokens	9.11	3.0
Std of Choice Tokens	2.36	0.52

Finally, in stage three, we construct additional data to mitigate the risk of memorization by large pretrained language models. In particular, for each question, we rephrase the original question using an open-source rephrasing tool without changing its answers or distractors.[5] This set is referred to as *Semantic Reconstruction*. Additionally, we leverage GPT-4 to reconstruct each question into a new context such that the misleading question premise is kept. In this case, both the question and the correct answer become different, but the reasoning path remains the same. After reconstruction, the distractors are collected in the same way as described earlier. This set is referred to as *Context Reconstruction*. A strong reasoning model is expected to solve all variants of the question consistently, as their reasoning patterns are identical despite being phrased differently. In total, we construct 1,119 data samples, including reconstruction variants. We report the key statistics in Table 1.

3 Task Description

3.1 Task Definition and Organization

In BRAINTEASER(S), we utilize both subtasks in the BRAINTEASER benchmark for evaluation: Sentenze Puzzle (*SP*) and Word Puzzle (*WP*). Both subtasks are multiple-choice QA tasks. We run our SemEval task on CodaLab. Our task is divided into two primary phases: (i) The Practice Phase runs from September 2023 to January 2024, and (ii) The Evaluation Phase runs from 10th Jan 2024 to 31st Jan 2024. We open the Post-Evaluation Phase after 31st Jan 2024 to encourage further research.

3.2 Evaluation Metrics and Data Splits

Evaluation Metrics We evaluate all systems using the same accuracy metrics as Jiang et al. (2023c):

[2] https://github.com/1171-jpg/BrainTeaser
[3] https://brainteasersem.github.io/
[4] https://codalab.lisn.upsaclay.fr/competitions/15566
[5] `https://quillbot.com/`

Table 2: Data statistics of each data split and baseline of BRAINTEASER(S).

	SP	WP
BRAINTEASER	627	492
Data Split of BRAINTEASER(S)		
Train	507	396
↪ Trial (*subset of train*)	120	96
Test	120	120
Baseline overall accuracy		
Human	0.920	0.917
ChatGPT (BRAINTEASER)	0.627	0.535
RoBERTa-L (BRAINTEASER)	0.434	0.207

Instance-based Accuracy considers each (original or reconstruction) question separately. We report instance-based accuracy on the original puzzles and their semantic and context reconstructions. *Group-based Accuracy* considers each original puzzle and its variants as a group. The model will score 1 only when it successfully solves all three puzzles in the group, otherwise, its score is 0. *Overall Accuracy* computes accuracy over all instances.

Data Split To enable BRAINTEASER(S) to support both fine-tuning and zero/few-shot setting, we further divided the original BRAINTEASER dataset into 3 data splits: train, trial, and test set, as shown in Table 2. The train set consists of 507 sentence puzzles and 396 word puzzles. We reuse a portion of the train set as a trial set, which contains 120 sentence puzzles and 96 word puzzles. The test set has 120 data for both subtasks. We release questions and answers from the train and trial set during the Practice Phase. We only release the questions of the test set during the Evaluation Phase and release the whole dataset after the Evaluation Phase ends.

Baseline We provide three baselines (Table 2, see Appendix A for details) to show the gap between humans and SOTA models. To get a comprehensive and robust evaluation performance for each subtask, the human evaluation is computed over 102 data randomly sampled from the original **BRAINTEASER** benchmark, ChatGPT and RoBERTa-L (Liu et al., 2019) performance are also computed over the **BRAINTEASER** in zero-shot setting, i.e. the original unpartitioned data of (Jiang et al., 2023c).

4 Participant System and Results

4.1 Participant Overview

We have 182 participants in total. In the Practice Phase, we have no limitation on the number of submissions to support exploration and enable participants to understand the submission format. We receive 243 submissions for *SP* and 155 for *WP*. In the Evaluation Phase, we allow up to three submissions per team and keep the submission with the best overall accuracy. Our final leaderboard has 48 team submissions for *SP* and 37 for *WP*.

4.2 Leaderboard Results

Table 3 (see Appendix A for full table) displays the top ten models for each subtask, ranked by overall accuracy. The best-performing model in *SP* excels in all six metrics, whereas the leading models in *WP* excel in all but context reconstruction. In the **instance-based accuracy metrics**, most top-performing models (75%) in two subtasks show better performance on original and semantic reconstruction compared to context reconstruction. Most models (80% in *SP*; 70% in *WP*) show the same trend across the entire leaderboard. In the **group-based accuracy metric**, half of the top models in both tasks align with their original instance-based accuracy for the grouped original and semantic reconstruction (Ori&Sem). Only one model in *WP* maintains its performance on all reconstructions (Ori&Sem&Con). Across the leaderboard, more than 80 percent of models in both subtasks show a decrease in Ori&Sem accuracy, ranging from 0.025 to 0.175 in *SP* and 0.031 to 0.281 in *WP*. Nearly all models show a significant drop in Ori&Sem&Con accuracy, with declines varying from 0.025 to 0.275 in *SP* and 0.031 to 0.344 in *WP*.

4.3 Fine-grained System Analysis

In this section, we provide system analysis for the models from the 28 system description papers from participants.*

Method Adaptation and Architecture Selection For both subtasks, the chosen adaptation methods among participants are either fine-tuning models (60%) or prompting models (65%) in a zero-shot (Sanh et al., 2021) or few-shot manner (Brown et al., 2020). Half of the participants try multiple adaptations and submit the best one. For the fine-tuning architecture, participants select either small-size models (<1B) including BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2020) or large-size models (>=1B) such as FLAN-T5 (Chung et al., 2022) and Mistral

* The rank discussed later in this section is based on systems with description papers.

Table 3: Top ten leaderboard results for both subtasks, including user submissions without system description papers. Ori = Original, Sem = Semantic, Con = Context. Team name with (*) submit the system description paper. The first, second and third submissions per category are represented by highlight, **bold** and underline, respectively.

Team Name	Overall	Instance-based			Group-based	
		Original	Semantic	Context	Ori & Sem	Ori & Sem & Con
Sentense Puzzle						
abdelhak*	0.983	1.000	1.000	0.950	1.000	0.950
HW-TSC*	**0.967**	1.000	**0.975**	**0.925**	**0.975**	0.900
Maxine	0.958	**0.975**	**0.975**	**0.925**	0.950	0.900
YingluLi	0.950	**0.975**	0.950	**0.925**	0.950	0.900
Theo	0.950	0.950	0.950	0.950	0.950	**0.925**
somethingx95	0.942	0.950	0.950	**0.925**	0.950	0.900
gerald	0.942	0.950	0.950	**0.925**	0.950	0.900
AmazUtah_NLP*	0.925	0.925	0.950	0.900	0.925	0.875
BITS Pilani*	0.900	**0.975**	0.925	0.800	0.925	0.775
ALF*	0.900	0.925	0.950	0.825	0.925	0.825
Word Puzzle						
Theo	0.990	1.000	1.000	**0.969**	1.000	0.969
gerald	0.990	1.000	1.000	**0.969**	1.000	0.969
somethingx95	**0.979**	1.000	1.000	0.938	1.000	**0.938**
zero_shot_is_all_you_need*	**0.979**	1.000	1.000	0.938	1.000	**0.938**
MasonTigers*	**0.979**	**0.969**	**0.969**	1.000	**0.969**	0.969
HW-TSC*	0.969	**0.969**	0.938	1.000	0.938	**0.938**
Maxine	0.969	**0.969**	0.938	1.000	0.938	**0.938**
YingluLi	0.969	**0.969**	0.938	1.000	0.938	**0.938**
kubapok	0.948	0.906	1.000	0.938	0.906	0.844
BITS Pilani*	0.917	0.938	0.938	0.875	0.938	0.812

7B (Jiang et al., 2023a). For the prompting architecture, the majority (90%) use closed-source LLMs such as GPT-4 (OpenAI et al., 2023), GPT-3.5, GeminiPro (Team et al., 2023), Claude (Anthropic, 2024), and Copilot.[6] Techniques like Chain-of-Thought (Wei et al., 2022a), Ensemble (Wang et al., 2022), and RECONCILE (Chen et al., 2023) are widely adopted for prompt engineering. Figure 2 provides a visualization of the overall accuracy distribution for each architecture. For fine-tuning architecture, fine-tuning on large models shows better performance with a tight accuracy range compared to small ones. Fine-tuning on small models shows competitive performance (three in the top five*) in *SP* but a significant drop in *WP*. Among the prompting designs, both zero-shot and few-shot show promising results (seven in the top nine systems*) on two subtasks, with the latter one having a wider accuracy range.

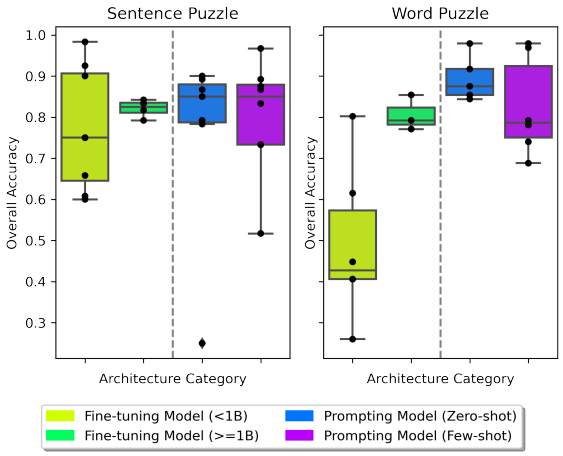


Figure 2: The overall accuracy distribution of each architecture selection.

External Dataset Half of the participants (54%) implement their systems only on the original target task, but some further introduce external datasets (35%) to enhance their models' performance. Participants generate humor-style synthetic data using LLMs, crawl riddle websites, or use RiddleSense (Lin et al., 2021) to invoke models' lateral thinking abilities. Other commonsense datasets such as BIRD-QA (Chen and Zulkernine, 2021) or knowledge graphs including ConceptNet (Speer et al., 2017) and WordNet (Miller, 1995) are used to provide general concepts of key instances in questions. Using humor-style datasets tends to be useful on both subtasks, especially for fine-tuning models. Meanwhile, synthetic explanations derived from LLMs are used in prompting to evoke chain-of-thought (Wei et al., 2022b) reasoning abilities.

Data Reconstruction Some participants (18%) reconstruct the original data or change the four-

[6]https://copilot.microsoft.com/

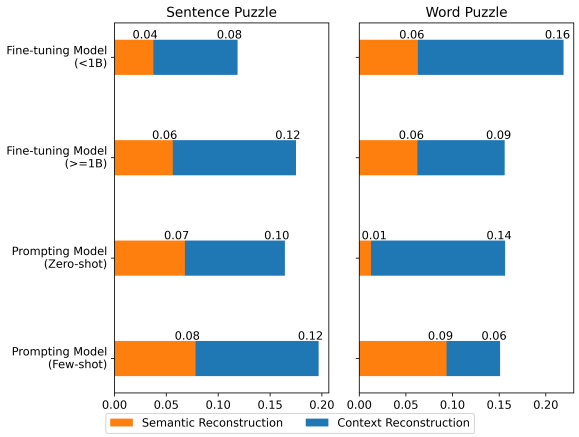


Figure 3: The drop in performance after introducing each reconstruction in group metric.

choice question format. Wang et al. (2024a) use back translation to enlarge the dataset size. Chakraborty et al. (2024) simplify each question into the binary choice problem and Reyes et al. (2024) solve the question under a classification approach with three class labels. Removing the unsure choice is also widely adopted for prompting, where the systems only choose unsure when they fail on the other three choices. Due to a limited number of data reconstruction samples, we cannot conclude which approach can improve performance.

Consistency of Model Predictions In Figure 3, we compare the drop in performance when considering reconstruction variants with group metrics to understand whether the models can solve lateral thinking puzzles by following a consistent reasoning path. On semantic reconstructions, the fine-tuning model has a smaller drop than zero/few-shot prompting in general. Fine-tuning on small models and zero-shot prompting work best on each subtask. On context reconstruction, all architectures show a more significant decline in performance. Fine-tuning on small models and few-shot prompting yield minimal drops in *SP* and *WP*, yet exhibit the largest declines in other subtasks.

5 Discussion

We start the discussion with the question: "*Is lateral thinking solved*?" The best-performing systems reach 100% on both tasks, making it seem that the task is solved. However, there remain many questions to explore. Our discussion targets 5 questions to provide overall insights: 1) What's the difference between the **BRAINTEASER(S)** SemEval Task and the original **BRAINTEASER** benchmark? 2) What's the difference between the best systems for sentence puzzles and word puzzles? 3) Are model predictions consistent with individual and group partitions? 4) What does fine-tuning mean for lateral thinking tasks? 5) What challenges still exist in the realm of lateral thinking?

5.1 Difference with the Original BRAINTEASER (Jiang et al., 2023c)

The **BRAINTEASER** benchmark (Jiang et al., 2023c) is proposed to evaluate LLMs' lateral thinking ability in **zero- and few-shot** settings while in **BRAINTEASER(S)** we release 80 percent of the data for training and we put no limitation on method adaptation. Although releasing data encourages more possibilities for participants, it also narrows down our hidden test set, making the comparison between system performance on BRAINTEASER(S) and the LLMs evaluation results on the BRAINTEASER benchmark unfair. With only 120 samples in the BRAINTEASER(S) test set, the probability of achieving high performance by some of the large number of systems becomes relatively large. Moreover, we expect that most of the lateral patterns will be recurring between the training and the test data, which especially benefits fine-tuning methods. With these caveats in mind, we hope the result and analysis on BRAINTEASER(S) can provide meaningful ideas and insight on lateral thinking and be verified systematically on the whole BRAINTEASER benchmark.

5.2 Effective System Choices and Differences

From subsection 4.3, we know architecture selection yields different distributions of performances on each subtask. On sentence puzzles, fine-tuning small models (Kelious and Okirim, 2024; Mishra and Ghashami, 2024; Farokh and Zeinali, 2024) with additional dataset providing competitive results. On word puzzles, either zero-shot (Moosavi Monazzah and Feghhi, 2024; Venkatesh and Sharma, 2024) or few-shot (Li et al., 2024; Raihan et al., 2024) prompting leads to top-performing results. In general, even small models obtaining language understanding during pre-training can adapt to sentence puzzles via fine-tuning, and additional humor-style datasets can evoke more lateral thinking abilities. On word puzzles, fine-tuned models have difficulties focusing on letter composition which hugely deviates from

their pertaining dataset. Even the top-scoring fine-tuning model (Kelious and Okirim, 2024) on *SP* fails to perform well on *WP*. On the other hand, the prompting method leverages the information stored in LLMs' parameters and their access to large pre-training data to mitigate the difficulty of word puzzles. However, the nature of the frozen model not only reduces the effectiveness of the external datasets but also limits further improvement and requires meticulous prompting engineering to ensure stable performance.

5.3 Prediction Consistency

Reconstruction of the original brainteaser puzzles allows us to distinguish between memorizing the training corpus and the ability of models to generalize to unseen samples. As indicated in subsection 4.2, most models struggle with consistent lateral thinking. Context reconstruction poses greater challenges than semantic reconstruction due to the need for lateral reasoning adaptation to novel settings. Context reconstruction of word puzzles is the most challenging, highlighting the risks of overfitting and memorization. Figure 3 shows architectures have different consistency issues. Fine-tuned models have a significant drop in context reconstruction in *WP* because the novelty of puzzles limits models to training corpus. Few-shot prompting can be beneficial for consistency in word puzzles but useless in sentence puzzles. LLMs' ability to follow pattern (Mirchandani et al., 2023) leads them to focus on the surface form in word puzzles, which brings improvement in consistency. Few-shot prompting can hardly provide general patterns of sentence puzzles due to its uniqueness, and the example in the demonstration can mislead the model.

5.4 Impact of Fine-Tuning

Even though recently in-context learning (ICL) (Brown et al., 2020) has achieved great progress on reasoning tasks (Talmor et al., 2019; Bisk et al., 2020), we are happy to see half of the participants implement their system in fine-tuning approaches and showing promising performance. Fine-tuning on small models can lead to a wide accuracy distribution, which requires careful design on hyperparameters and the training process. Exposure to external datasets can stabilize and enhance performance. Fine-tuning on large models shows tight accuracy distribution but lacks top-performing models, which suggests the need for more fine-tuning data to "distort" the default commonsense (Kumar et al., 2022) and evoke lateral thinking out-of-distribution (Jiang et al., 2023b). Also, the large gap between instance- and group-based metric (Figure 3) points out that short-cut learning still exists among these methods.

5.5 Challenges in Lateral Thinking

We summarize the discussion with the challenges that remain unsolved and require further effort to evoke the models' lateral thinking abilities. 1) The system performances and our analysis are based on a small set of original BRAINTEASER benchmark (subsection 5.1). A more general and systematic analysis should be performed with the entire original BRAINTEASER data or even an enlarged version of it, starting from prompting models. 2) There is still a lack of a general approach demonstrating a stable and competitive performance on both subtasks. No existing method can merge the advantages of each architecture on each subtask (subsection 5.2). 3) Each model fails to generate consistent predictions similar to humans, even under simple semantic reconstructions (subsection 5.3). 4) Fine-tuning methods suffer from learning shortcuts while prompting methods have problems finding general lateral thinking patterns akin to humans (see also (Lewis and Mitchell, 2024)) (subsection 5.4).

6 Conclusions and Future Perspectives

This paper summarizes SemEval 2024 Task 9, BRAINTEASER(S), a novel task defying common sense. We present the motivation, data design, data construction, evaluation process, competition systems, participant results, result analysis, and discussion. BRAINTEASER(S) was popular among participants and received 483 submissions from 182 teams during the competition, with various method adaptations and architecture selections demonstrating different advantages on each subtask and evaluation metric. The best-performing systems have impressive performance on both subtasks, which reach 100% accuracy on lateral thinking puzzles from the web. However, our fine-grained analysis highlights the remaining questions and challenges for further research. Importantly, BRAINTESER(S) SemEval result is evaluated over a subset (20%) of original BRAINTEASER benchmark. Even on this subset and despite the access to 80% of the data for training, models still strug-

gle to reason consistently on semantic and context reconstruction. Future work should investigate flexible ways to combine lateral and vertical thinking, construct better evaluation metrics for creative and open-ended generations, build connections within reconstruction based on analogical reasoning (Sourati et al., 2023) and explore a dynamic, multi-stage process where the model (or human) can request clarifications or obtain contextual hints. The BRAINTEASER(S) SemEval Task, together with its source BRAINTEASER task, is the first step toward injecting AI systems with lateral thinking ability. We hope that the competition results and analysis can inspire future research on developing and evaluating lateral thinking models.

Ethical Considerations

As our brain teasers are "folk knowledge" and are published on a range set of websites, it is hard to check their original licenses comprehensively. Yet, the website owners declare permission to print and download material for **non-commercial use** without modification on the material's copyright. Therefore, we provide the corresponding copyright statements and website URLs for each original brain teaser and its adversarial version. In addition, we ask the task participants to sign a document claiming that the only aim of the data usage is research. We note that, despite our best efforts, the task data may still contain bias in terms of gender or politics. We will indicate that future research should use the task data with caution.

7 Acknowledgements

We appreciate Baktash Ansari, Dilip Venkatesh, Soumya Smruti Mishra, Harshit Gupta, and Pouya Sadeghi for their support as emergency reviewers for the competition. This research was sponsored by the Defense Advanced Research Projects Agency via Contract HR00112390061, Defense Advanced Research Projects Agency with award N660011924033 and Strengthening Teamwork for Robust Operations in Novel Groups via number W911NF-19-S-0001.

References

Mohammad Hossein Abbaspour, Erfan Moosavi Monazzah, and Sauleh Eetemadi. 2024. Iust-nlplab at semeval-2024 task 9: Brainteaser by mpnet (sentence puzzle). In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1095–1098, Mexico City, Mexico. Association for Computational Linguistics.

Baktash Ansari, Mohammadmostafa Rostamkhani, and Sauleh Eetemadi. 2024. Bamo at semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 224–232, Mexico City, Mexico. Association for Computational Linguistics.

Anthropic. 2024. Introducing claude 2.1.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, 05, pages 7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Trina Chakraborty, Marufur Rahman, and Md Omar Faruqe. 2024. Deja vu at semeval 2024 task 9: A comparative study of advanced language models for commonsense reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1229–1234, Mexico City, Mexico. Association for Computational Linguistics.

Alvin Chen, Ray Groshan, and Sean Von Bayern. 2024. Mothman at semeval-2024 task 9: An iterative system for chain-of-thought prompt optimization. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1888–1900, Mexico City, Mexico. Association for Computational Linguistics.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *arXiv preprint arXiv:2309.13007*.

Yuhao Chen and Farhana Zulkernine. 2021. Bird-qa: a bert-based information retrieval approach to domain specific question answering. In *2021 IEEE International Conference On Big Data (Big Data)*, pages 3503–3510. IEEE.

Kyu Hyun Choi and Seung-Hoon Na. 2024. Geminipro at semeval-2024 task 9: Brainteaser on gemini. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1626–1630, Mexico City, Mexico. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Seyed Ali Farokh and Hossein Zeinali. 2024. Alf at semeval-2024 task 9: Exploring lateral thinking capabilities of lms through multi-task fine-tuning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1534–1539, Mexico City, Mexico. Association for Computational Linguistics.

Harshit Gupta, Manav Chaudhary, Shivansh Subramanian, Tathagata Raha, and Vasudeva Varma. 2024. irel at semeval-2024 task 9: Improving conventional prompting methods for brain teasers. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1769–1777, Mexico City, Mexico. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Ethan Heavey, James Hughes, and Milton King. 2024. Stfx-nlp at semeval-2024 task 9: Brainteaser: Three unsupervised riddle-solvers. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 28–33, Mexico City, Mexico. Association for Computational Linguistics.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2023b. Transferring procedural knowledge across commonsense tasks. *arXiv preprint arXiv:2304.13867*.

Yifan Jiang, Filip Ilievski, Kaixin Ma, and Zhivar Sourati. 2023c. BRAINTEASER: Lateral thinking puzzles for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14317–14332, Singapore. Association for Computational Linguistics.

Abdelhak Kelious and Mounir Okirim. 2024. Abdelhak at semeval-2024 task 9 : Decoding brainteasers, the efficacy of dedicated models versus chatgpt. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 200–205, Mexico City, Mexico. Association for Computational Linguistics.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*.

Martha Lewis and Melanie Mitchell. 2024. Using counterfactual tasks to evaluate the generality of analogical reasoning in large language models. *arXiv preprint arXiv:2402.08955*.

Yinglu Li, Zhao Yanqing, Min Zhang, Yadong Deng, Aiju Geng, Xiaoqin Liu, Mengxin Ren, Yuang Li, Su Chang, Xiaofeng Zhao, Xiaosong Qiao, Ming Zhu, Yilun Liu, Mengyao Piao, Feiyu Yao, shimin tao, Hao Yang, and Yanfei Jiang. 2024. Hw-tsc at semeval-2024 task 9: Exploring prompt engineering strategies for brain teaser puzzles through llms. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1657–1662, Mexico City, Mexico. Association for Computational Linguistics.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. *arXiv preprint arXiv:2101.00376*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021a. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 15, pages 13507–13515.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Satoru Ozaki, Eric Nyberg, and Alessandro Oltramari. 2021b. Exploring strategies for generalizable commonsense reasoning with pre-trained models. *EMNLP 2021*.

Suyash Vardhan Mathur, Akshett Jindal, and Manish Shrivastava. 2024. Davinci at semeval-2024 task 9: Few-shot prompting gpt-3.5 for unconventional reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1202–1206, Mexico City, Mexico. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Suvir Mirchandani, Fei Xia, Pete Florence, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, Andy Zeng, et al. 2023. Large language models as general pattern machines. In *7th Annual Conference on Robot Learning*.

Soumya Mishra and Mina Ghashami. 2024. Amazutah_nlp at semeval-2024 task 9: A multichoice question answering system for commonsense defying reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1447–1453, Mexico City, Mexico. Association for Computational Linguistics.

Erfan Moosavi Monazzah and Mahdi Feghhi. 2024. Zero shot is all you need at semeval-2024 task 9: A study of state of the art llms on lateral thinking puzzles. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1901–1905, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, and etc Lama Ahmad. 2023. Gpt-4 technical report.

Ioannis Panagiotopoulos, George Filandrianos, Maria Lymperaiou, and Giorgos Stamou. 2024. Ails-ntua at semeval-2024 task 9: Cracking brain teasers: Transformer models for lateral thinking puzzles. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1744–1757, Mexico City, Mexico. Association for Computational Linguistics.

Zahra Rahimi, Mohammad Moein Shirzady, Zeinab Taghavi, and Hossein Sameti. 2024. Nimz at semeval-2024 task 9: Evaluating methods in solving brainteasers defying commonsense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 148–154, Mexico City, Mexico. Association for Computational Linguistics.

Md Nishat Raihan, Dhiman Goswami, Al Nahian Bin Emran, Sadiya Sayara Chowdhury Puspo, Amrita Ganguly, and Marcos Zampieri. 2024. Masontigers at semeval-2024 task 9: Solving puzzles with an ensemble of chain-of-thought prompts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1360–1365, Mexico City, Mexico. Association for Computational Linguistics.

Cecilia Reyes, Orlando Ramos-Flores, and Diego Martínez-Maqueda. 2024. Iimas at semeval-2024 task 9: A comparative approach for brainteaser solutions. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1110–1115, Mexico City, Mexico. Association for Computational Linguistics.

Mohammadmostafa Rostamkhani, Shayan Mousavinia, and Sauleh Eetemadi. 2024. Rosha at semeval-2024 task 9: Brainteaser: A novel task defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1027–1031, Mexico City, Mexico. Association for Computational Linguistics.

Pouya Sadeghi, Amirhossein Abaskohi, and Yadollah Yaghoobzadeh. 2024. utebc-nlp at semeval-2024 task 9: Can llms be lateral thinkers? In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1778–1789, Mexico City, Mexico. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *AAAI Conference on Artificial Intelligence*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

Vineet Saravanan and Steven Wilson. 2024. Ounlp at semeval-2024 task 9: Retrieval-augmented generation for solving brain teasers with llms. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 206–212, Mexico City, Mexico. Association for Computational Linguistics.

Marco Siino. 2024. Deberta at semeval-2024 task 9: Using deberta for defying common sense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 290–296, Mexico City, Mexico. Association for Computational Linguistics.

Zhivar Sourati, Filip Ilievski, Pia Sommerauer, and Yifan Jiang. 2023. Arn: A comprehensive framework and benchmark for analogical reasoning on narratives.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, 1.

Kejsi Take and Chau Tran. 2024. Riddlemasters at semeval-2024 task 9: Comparing instruction finetuning with zero-shot approaches. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1393–1398, Mexico City, Mexico. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL-HLT*, pages 4149–4158.

Gemini Team, Rohan Anil, Sebastian Borgeaud, and etc Yonghui Wu. 2023. Gemini: A family of highly capable multimodal models.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the seventh workshop on building educational applications using NLP*, pages 163–173.

Dilip Venkatesh and Yashvardhan Sharma. 2024. Bits pilani at semeval-2024 task 9: Prompt engineering with gpt-4 for solving brainteasers. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 803–807, Mexico City, Mexico. Association for Computational Linguistics.

Jie Wang, Jin Wang, and Xuejie Zhang. 2024a. Ynuhpcc at semeval-2024 task 9: Using pre-trained language models with lora for multiple-choice answering tasks. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 458–463, Mexico City, Mexico. Association for Computational Linguistics.

Weiqi Wang, Baixuan Xu, Haochen Shi, Jiaxin Bai, Qi Hu, and Yangqiu Song. 2024b. Knowcomp at semeval-2024 task 9: Conceptualization-augmented prompting with large language models for lateral reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1650–1656, Mexico City, Mexico. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022a. Chain-of-thought prompting elicits reasoning in large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Qi Yang, Jingjie Zeng, Liang Yang, and Hongfei Lin. 2024. yangqi at semeval-2024 task 9: Simulate human thinking by large language model for lateral thinking challenges. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 233–238, Mexico City, Mexico. Association for Computational Linguistics.

Jiarui Zhang, Filip Ilievski, Kaixin Ma, Jonathan Francis, and Alessandro Oltramari. 2022. A study of zero-shot adaptation with commonsense knowledge. *Automated Knowledge Base Construction(AKBC)*.

Micah Zhang, Shafiuddin Rehan Ahmed, and James H. Martin. 2024. Ftg-cot at semeval-2024 task 9: Solving sentence puzzles using fine-tuned language models and zero-shot cot prompting. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1235–1241, Mexico City, Mexico. Association for Computational Linguistics.

A CodaLab Leaderboard

In the main part of the paper, we only analyse the results for part of the participants' submission due to page limitation. Table 4 and 5 show a complete set of user names and results of the participants in the CodaLab competition for two subtasks, including users who did not submit a system description. The human evaluation is computed over 102 data randomly sampled from the **whole dataset**. The random base is average over three different seeds. The ChatGPT and RoBERTa-L baseline is computed over the whole dataset using OPENAI API[7] from 2023/5/01 to 2023/5/15.

We visualize each team's overall accuracy in each subtask according to the model adaptation category in Figure 4. In Sentence Puzzle, 12 teams employed fine-tuning, and 15 adopted zero/few-shot approaches. Fine-tuning achieved 1st, 3rd, and 5th positions on the leaderboard, whereas zero/few-shot have 7 places in the top ten. For Word Puzzle, 9 teams used fine-tuning, and 11 opted for zero/few-shot, with the latter dominating the top five ranks, outperforming fine-tuning.

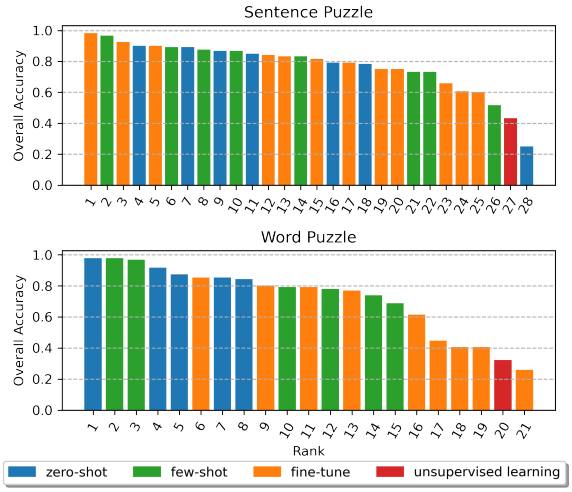


Figure 4: The overall accuracy performance of each team based on method adaptations.

B Participant Systems

In this section, we list the systems of all participants who submitted a system description paper. The **team name** represents each system, appended with the corresponding rank in [bracket], keywords in (parentheses), and a short description for further reference. *SP X* and*WP X* represent the ranks in sentence and word puzzles based on overall performance, respectively.

Abdelhak [SP 1;WP 16] (Kelious and Okirim, 2024) (*Fine-tuned*;*DeBERTa*;*Zero-shot*;*ChatGPT*;*Temperature Anlysis*) They fine-tuned the pre-trained language model DeBERTa-v3-base in the multiple-choice setting. They further experimented with the relationship between temperature and lateral thinking with ChatGPT in a zero-shot setting.

HW-TSC [SP 2;WP 3] (Li et al., 2024)(*Fine-tuned*;*Mixstral*;*Zero-shot*;*Few-shot*;*GPT-3.5*;*GPT-4*;*Prompting Engineering*;*Ensemble*) They first experimented with fine-tuning Mixtral overall whole training set. They turned to GPT-3.5 and GPT-4 due to poor fine-tuning results. They identified and categorized over 20 challenging training instances to include in an extended prompt. Finally, they submitted their result with GPT-4 in the few-shot setting with a well-designed prompting demonstration as well as the ensemble method.

AmazUtah_NLP [SP 6;WP 10] (Mishra and Ghashami, 2024) (*Fine-tuned*;*DeBERTa*;*BERT*;*External Data*;*Synthetic Data*;*RiddleSense*) They fine-tuned DeBERTa and BERT in the multiple-choice setting. They utilized the public puzzle dataset RiddleSense as well as creating humor-style data by prompting GPT 4 as the external dataset. They also experimented by adding commonsense datasets SWAG and CODAH but found the introduction reduced overall performance.

BITS Pilani [SP 7;WP 5] (Venkatesh and Sharma, 2024) (*Zero-shot*;*GPT-4*;*Prompting Engineering*) They used OpenAI's GPT-4 model along with prompt engineering in the zero-shot setting to solve these brainteasers.

ALF [SP 7] (Farokh and Zeinali, 2024) (*Fine-tuned*;*ALBERT*;*RoBERTa*;*DeBERTa*;*Flan T5*;*Unified QA*;*External Data*;*RiddleSense*) Their experiments focused on two prominent families of pre-trained models, BERT and T5, and fine-tuned ALBERT, RoBERTa, DeBERTa, Flan T5 and Unified QA in the multiple-choice setting. They explored the potential benefits of multi-task finetuning on commonsense reasoning datasets, including RiddleSense, CSQA, PIQA, SIQA, Hellaswag, and SWAG, to enhance performance.

uTeBC-NLP [SP 8] (Sadeghi et al., 2024) (*Fine-tuned*;*Zephyr-7B-β*;*Zero-shot*;*Few-shot*;*GPT-3.5*;*GPT-*

[7]https://platform.openai.com/docs/api-reference

4;*RAG*;*External Data*;*Synthetic Data*;*Prompting Engineering*;*COT*;*Lateral thinking enhancement analysis*) They explored Chain of Thought (CoT) strategies, enhancing prompts with detailed task descriptions, and retrieval augmented generation for generating in-context samples. Their experiments involve GPT-3.5 and GPT-4. They also showcased that fine-tuning Zephyr-7B-β with a lateral thinking approach significantly enhances the model's performance on other commonsense datasets.

yangqqi [SP 8;WP 6] (Yang et al., 2024) (*Zero-shot*;*ChatGPT*;*RAG*;*Self-Adaptive ICL*;*Prompting Engineering*;*External Data*;*ConceptNet*) They proposed the SHTL system to mimic human lateral thinking ability for solving brain teaser questions. They first retrieved related knowledge concepts from ConceptNet and used SAICL to find the optimal organization for each single test sample. At last, they provide ChatGPT with the related knowledge concepts and find the options to solve the conflicts contained in the related knowledge concepts effectively.

Mothman [SP 9] (Chen et al., 2024) (*Zero-shot*;*Few-shot*;*GPT-4*;*Prompting Engineering*;*COT*;) They proposed a system for iterative chain-of-thought prompt engineering which optimizes prompts using a flexible evaluation strategy on both model outputs and input data. They obtain feedback from human evaluation to modify the prompting demonstration interactively to guide GPT-4 to focus on challenging problems. They also proposed a new COT strategy requiring GPT-4 to produce rationals for both correct and incorrect options.

Zero_Shot_is_All_You_Need [SP 10;WP 2] (Moosavi Monazzah and Feghhi, 2024) (*Zero-shot*;*Bing*;*Gemini*;*Mixtral*;*Mixtral*;*ChatGPT*;*Phi-2*;*Prompting Engineering*;*Ensemble*;*Debate*) They examined the zero-shot ability of current state-of-the-art LLMs, Bing, Gemini, Mixtral, ChatGPT and Phi-2 to solve this task. They also tried ensemble and debate prompting engineering methods.

OUNLP [SP 10;WP 11] (Saravanan and Wilson, 2024) (*Zero-shot*;*Few-shot*;*GPT-3.5*;*GPT-4*;*Gemini*;*languagemodels*;*Prompting Engineering*;*COT*;*RECONCILE*;*External Data*;*crawled riddles*) They experimented with a series of structured prompts ranging from basic to those integrating task descriptions and explanations(COT). They use the most similar or the most different training example as the demonstration in the one-shot prompting. They downloaded a collection of riddles from the web as an external data source. In the end, they simulated a council scenario to evoke discussion between different models but didn't observe significant improvement.

BAMO [SP 11] (Ansari et al., 2024) (*Fine-tuned*;*RoBERTa*;*BERT*;*Zero-shot*;*Open Chat*;*Llama-2-70b*;*Mixtral*;*GPT3.5*;*Claud*;*Microsoft Copilot*;*Prompting Engineering*;*ReConcile*) They fine-tuned 2 models, BERT and RoBERTa Large, and employed a Chain of Thought (CoT) zero-shot prompting approach with 6 large language models, such as GPT-3.5, Mixtral, and Llama2. Finally, they utilized ReConcile prompting amount three models.

YNU-HPCC [SP 12;WP 13] (Wang et al., 2024a) (*Fine-tuned*;*DeBERTa*;*External Data*;*Back translation*) They fine-tuned DeBERTa in different training strategies and enhanced the training set with back translation.

FtG-CoT [SP 13] (Zhang et al., 2024) (*Fine-tuned*;*BERT*;*Zero-shot*;*Few-shot*;*GPT-3.5*;*Prompting Engineering*;*COT*) They first fine-tuned BERT in a multi-class classification setting and fine-tuned GPT-3.5 with chain-of-thought generated by zero-shot prompting. Then they picked the set of training demonstrations provided in the few-shot prompt based on the BERT encoding cosine similarity to the test question.

MasonTigers [SP 13;WP 2] (Raihan et al., 2024) (*Zero-shot*;*Few-shot*;*GPT-4.5*;*Claude*;*Mixtral*;*Prompting Engineering*;*COT*) They explored various prompting strategies to guide the models, including zero-shot, few-shot, and chain-of-thought prompting. The Ensemble method was adopted to enhance COT performance.

AILS-NTUA [SP 14;WP 7] (Panagiotopoulos et al., 2024) (*Fine-tuned*;*DeBERTa*;*RoBERTa*;*BERT*;*Mixtral*;*Llama 2*;*Phi-2*) They evaluated a plethora of pre-trained transformer-based language models of different sizes and pre-train dataset through fine-tuning. They also delved into models' frequent failures to obtain a deeper understanding of reasoning cues that make models struggle the most.

RiddleMaster [SP 15;WP 8] (Take and Tran, 2024) (*Fine-tuned*;*Mistral*;*Zero-shot*;*GPT-4*;*Prompting Engineering*;*COT*;*Ensemble*) They compared multiple zero-shot approaches using

GPT-4 as well as fine-tuned Mistral output.

UMBCLU[8] [SP 15;WP 11] (*Fine-tuned*;*Flan-T5*;*Data Augmentation*) They fine-tuned and evaluated various T5 family models on both the word and sentence puzzle tasks and showed that training on the alternative contexts improves a model's lateral reasoning capability.

KnowComp [SP 16;WP 7] (Wang et al., 2024b) (*Zero-shot*;*ChatGPT*;*Prompting Engineering*) They first prompted ChatGPT to identify relevant instances in the question and generate conceptualizations for the identified instances. They then converted each puzzle into a declarative format and modified the task to involve selecting the most plausible statement from the options.

NIMZ [SP 20;WP 19] (Rahimi et al., 2024) (*Fine-tuned*;*BERT*;*RoBERTa*;*T5*;*QA-GNN*;*External Data*;*ConceptNet*) They fine-tuned BERT, RoBERTa and T5 and evaluated their performance. They used ConceptNet as an external knowledge source and fine-tuned graph neural network QA-GNN and suggested its superiority on sentence puzzle.

Deja-Vu [SP 20;WP 20] (Chakraborty et al., 2024) (*Fine-tuned*;*BERT*;*RoBERTa*;*XLNet*;*BART*;*T5*;*Data Augmentation*) They fine-tuned five transformer-based language models and found the integration of sentence and word puzzles into a single dataset led to a noticeable decrease in accuracy.

GeminiPro [SP 21;WP 12] (Choi and Na, 2024) (*Zero-shot*;*Few-shot*;*Gemini*;*Prompting Engineering*) They tested Gemini's performance in zero-shot and few-shot settings. They experimented with whether tailor-made demonstrations to specific tasks can alleviate confusion and aid in 049 problem-solving.

iREL [SP 21;WP 14] (Gupta et al., 2024) (*Zero-shot*;*Few-shot*;*Gemini*;*Prompting Engineering*;*COT*) They tested Gemini's performance in zero-shot and few-shot settings. Especially in the few-shot setting, reasoning from Gemini and GPT-4 are integrated into the demonstration, selected by static or dynamic strategy.

IIMAS [SP 23;WP 22] (Reyes et al., 2024) (*Fine-tuned*;*BERT*;*RoBERTa*;*ChatGPT*;*Gemini*;*Data Augmentation*) They tackled this challenge by applying fine-tuning techniques with pre-trained models (BERT and RoBERTa Winogrande) while also augmenting the dataset with the LLMs ChatGPT and Gemini. During the training, they transformed the data format for specific templates.

IUST-NLPLAB [SP 24] (Abbaspour et al., 2024) (*Fine-tuned*;*MPNET*;*Zero-shot*;*GPT-3.5*) They first introduced a zero-shot approach leveraging the capabilities of the GPT3.5 model. Additionally, they presented three finetuning methodologies utilizing MPNET as the underlying architecture, each employing a different loss function.

ROSHA [SP 25;WP 20] (Rostamkhani et al., 2024) (*Fine-tuned*;*RoBERTa*;*Zero-shot*;*GPT-3.5*;*Gemini*;*Mixtral*;*GPT-4*;*External Data*;*BiRdQA*;*RiddleSense*;*Prompting Engineering*;*Reconcile*) They applied the XLM-RoBERTa model both to the original training dataset and concurrently to the original dataset alongside the BiRdQA dataset and the RiddleSense for comprehensive model training. They also tested the Reconcile prompting strategy with GPT-3.5, Gemini as well as Mixtral and zero-shot on GPT-4.

DaVinci [SP 26;WP 15] (Mathur et al., 2024) (*Few-shot*;*GPT-3.5*;*Prompting Engineering*) They used few-shot prompting on GPT-3.5 with rationale and gained insights regarding the difference in the nature of the two types of questions.

StFX-NLP [SP 27;WP 21] (Heavey et al., 2024) (*unsupervised*;*External Data*;*WordNet*) They explored three unsupervised learning models. Two of these models incorporate word sense disambiguation and part-of-speech tagging, specifically leveraging SensEmBERT and the Stanford log-linear part-of-speech tagger. The third model relies on a more traditional language modelling approach.

DeBERTa [SP 28] (Siino, 2024) (*Zero-shot*;*DeBERTa*) They used DeBERTa in zero-shot setting.

[8]The paper was withdrawn.

Table 4: Oveview of results of Sentence-puzzle subtask, including user submissions without system description papers. Ori = Original, Sem = Semantic, Con = Context. Team name with (*) submitted the system description paper. The first, second and third submissions per category are represented by highlight, **bold** and underline, respectively.

Team Name	Overall	Instance-based			Group-based	
		Original	Semantic	Context	Ori & Sem	Ori & Sem & Con
Abdelhak*	0.983	1.000	1.000	0.950	1.000	0.950
HW-TSC*	**0.967**	1.000	**0.975**	**0.925**	**0.975**	0.900
Maxine	0.958	**0.975**	**0.975**	**0.925**	0.950	0.900
YingluLi	0.950	**0.975**	0.950	**0.925**	0.950	0.900
Theo	0.950	0.950	0.950	0.950	0.950	**0.925**
somethingx95	0.942	0.950	0.950	**0.925**	0.950	0.900
gerald	0.942	0.950	0.950	**0.925**	0.950	0.900
AmazUtah_NLP*	0.925	0.925	0.950	0.900	0.925	0.875
BITS Pilani*	0.900	**0.975**	0.925	0.800	0.925	0.775
ALF*	0.900	0.925	0.950	0.825	0.925	0.825
uTeBC-NLP*	0.892	**0.975**	0.875	0.825	0.850	0.750
jkarolczak	0.892	**0.975**	0.875	0.825	0.875	0.775
kubapok	0.892	0.925	0.900	0.850	0.900	0.825
yangqi*	0.892	0.900	0.900	0.875	0.900	0.875
Mothman*	0.875	**0.975**	0.850	0.800	0.850	0.700
zero_shot_is_all_you_need*	0.867	0.950	0.825	0.825	0.800	0.725
OUNLP*	0.867	0.950	0.875	0.775	0.850	0.725
justingu	0.850	0.950	0.825	0.775	0.825	0.700
BAMO*	0.850	0.900	0.825	0.825	0.825	0.700
YNU-HPCC*	0.842	0.900	0.825	0.800	0.825	0.725
FtG-CoT*	0.833	0.900	0.825	0.775	0.800	0.675
MasonTigers*	0.833	0.850	0.825	0.825	0.800	0.700
AILS-NTUA*	0.817	0.850	0.825	0.775	0.825	0.700
RiddleMaster*	0.792	0.800	0.775	0.800	0.725	0.650
UMBCLU*	0.792	0.750	0.850	0.775	0.725	0.600
johnp	0.783	0.850	0.775	0.725	0.750	0.675
MABUSETTEH	0.783	0.800	0.775	0.775	0.775	0.700
KnowComp*	0.783	0.825	0.775	0.750	0.725	0.625
ehsan.tavan	0.775	0.800	0.800	0.725	0.775	0.675
amr8ta	0.775	0.775	0.775	0.775	0.750	0.650
yiannispn	0.767	0.800	0.800	0.700	0.750	0.625
haha123	0.758	0.825	0.775	0.675	0.750	0.625
adriti	0.758	0.750	0.725	0.800	0.725	0.675
TienDat23	0.758	0.725	0.800	0.750	0.675	0.525
Deja_Vu*	0.750	0.775	0.700	0.775	0.700	0.625
NIMZ*	0.750	0.750	0.725	0.775	0.700	0.675
iREL*	0.733	0.775	0.725	0.700	0.700	0.575
GeminiPro*	0.733	0.750	0.750	0.700	0.700	0.600
caoyongwang	0.725	0.800	0.700	0.675	0.700	0.550
IIMAS*	0.658	0.650	0.675	0.650	0.600	0.500
IUST-NLPLAB*	0.608	0.625	0.625	0.575	0.625	0.500
ROSHA*	0.600	0.625	0.575	0.600	0.500	0.375
Team DaVinci*	0.517	0.575	0.550	0.425	0.500	0.300
StFX-NLP*	0.433	0.425	0.400	0.475	0.350	0.200
Team 9	0.250	0.275	0.275	0.200	0.100	0.000
DeBERTa*	0.250	0.225	0.250	0.275	0.200	0.075
amirhallaji	0.242	0.225	0.200	0.300	0.050	0.025
maryam.najafi	0.233	0.225	0.275	0.200	0.100	0.025
Human (Jiang et al., 2023c)	0.920	0.907	0.907	0.944	0.907	0.889
GPT-4 (BREAINTEASER)	0.898	0.942	0.900	0.852	0.880	0.775
GPT-4 (BREAINTEASER(S))	0.858	0.925	0.825	0.825	0.8	0.775
ChatGPT (BREAINTEASER)	0.627	0.608	0.593	0.679	0.507	0.397
RoBERTa-L (BREAINTEASER)	0.434	0.435	0.402	0.464	0.330	0.201
Random	0.244	0.255	0.249	0.228	0.056	0.014

Table 5: Oveview of results of Word-puzzle subtask, including user submissions without system description papers. Ori = Original, Sem = Semantic, Con = Context. Team name with (*) submitted the system description paper. The first, second and third submissions per category are represented by highlight, **bold** and underline, respectively.

Team Name	**Overall**	Instance-based			Group-based	
		Original	**Semantic**	**Context**	**Ori & Sem**	**Ori & Sem & Con**
Theo	0.990	1.000	1.000	**0.969**	1.000	0.969
gerald	0.990	1.000	1.000	**0.969**	1.000	0.969
somethingx95	**0.979**	1.000	1.000	0.938	1.000	**0.938**
zero_shot_is_all_you_need*	**0.979**	1.000	1.000	0.938	1.000	**0.938**
MasonTigers*	**0.979**	**0.969**	**0.969**	1.000	**0.969**	0.969
HW-TSC*	0.969	**0.969**	0.938	1.000	0.938	**0.938**
Maxine	0.969	**0.969**	0.938	1.000	0.938	**0.938**
YingluLi	0.969	**0.969**	0.938	1.000	0.938	**0.938**
kubapok	0.948	0.906	1.000	0.938	0.906	0.844
BITS Pilani*	0.917	0.938	0.938	0.875	0.938	0.812
justingu	0.917	0.938	0.938	0.875	0.906	0.781
jkarolczak	0.875	0.906	0.938	0.781	0.875	0.688
yangqi*	0.875	0.906	0.938	0.781	0.906	0.688
ehsan.tavan	0.875	0.906	0.875	0.844	0.812	0.750
AILS-NTUA*	0.854	0.875	0.906	0.781	0.812	0.719
johnp	0.854	0.875	0.906	0.781	0.812	0.719
caoyongwang	0.854	0.844	0.844	0.875	0.781	0.719
KnowComp*	0.854	0.844	0.906	0.812	0.844	0.656
RiddleMaster*	0.844	0.844	0.844	0.844	0.781	0.656
yiannispn	0.833	0.844	0.844	0.812	0.719	0.625
AmazUtah_NLP*	0.802	0.844	0.812	0.750	0.781	0.594
OUNLP*	0.792	0.781	0.812	0.781	0.719	0.531
UMBCLU*	0.792	0.781	0.750	0.844	0.719	0.625
TienDat23	0.792	0.844	0.750	0.781	0.750	0.625
GeminiPro*	0.781	0.781	0.719	0.844	0.594	0.594
YNU-HPCC*	0.771	0.781	0.719	0.812	0.719	0.625
iREL*	0.740	0.719	0.719	0.781	0.562	0.531
Team DaVinci*	0.688	0.719	0.719	0.625	0.594	0.469
Abdelhak*	0.615	0.625	0.625	0.594	0.562	0.406
amr8ta	0.604	0.625	0.625	0.562	0.594	0.438
adriti	0.604	0.656	0.625	0.531	0.625	0.375
MABUSETTEH	0.583	0.594	0.625	0.531	0.562	0.281
NIMZ*	0.448	0.438	0.469	0.438	0.406	0.219
Deja_Vu*	0.406	0.375	0.469	0.375	0.344	0.125
ROSHA*	0.406	0.438	0.375	0.406	0.375	0.250
StFX-NLP*	0.323	0.406	0.219	0.344	0.125	0.062
IIMAS*	0.260	0.250	0.250	0.281	0.125	0.062
Human (Jiang et al., 2023c)	0.917	0.917	0.917	0.917	0.917	0.896
GPT-4 (BREAINTEASER)	0.736	0.811	0.756	0.640	0.689	0.494
GPT-4 (BREAINTEASER(S))	0.854	0.875	0.875	0.813	0.781	0.625
ChatGPT (BREAINTEASER)	0.535	0.561	0.524	0.518	0.439	0.293
RoBERTa-L (BREAINTEASER)	0.207	0.195	0.195	0.232	0.146	0.061
Random	0.260	0.279	0.225	0.073	0.018	0.253

SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes

Dimitar Dimitrov[1], **Firoj Alam**[2], **Maram Hasanain**[2], **Abul Hasnat**[34], **Fabrizio Silvestri**[5], **Preslav Nakov**[6] and **Giovanni Da San Martino**[7]
[1]Sofia University "St. Kliment Ohridski", [2]Qatar Computing Research Institute, HBKU, Qatar
[3]APAVI.AI, France, [4]BlackBird.AI, USA, [5]Sapienza University of Rome, Italy,
[6]Mohamed bin Zayed University of Artificial Intelligence, UAE
[7]Department of Mathematics, University of Padova, Italy
ilijanovd@fmi.uni-sofia.bg, {fialam,mhasanain}@hbku.edu.qa
fsilvestri@diag.uniroma1.it,hasnat@blackbird.ai
preslav.nakov@mbzuai.ac.ae, dasan@math.unipd.it

Abstract

The automatic identification of misleading and persuasive content has emerged as a significant issue among various stakeholders, including social media platforms, policymakers, and the broader society. To tackle this issue within the context of memes, we organized a shared task at SemEval-2024, focusing on the multilingual detection of persuasion techniques. This paper outlines the dataset, the organization of the task, the evaluation framework, and the outcomes.The task targets memes in four languages, with the inclusion of three surprise test datasets in Bulgarian, North Macedonian, and Arabic. It encompasses three subtasks: *(i)* identifying whether a meme utilizes a persuasion technique; *(ii)* identifying persuasion techniques within the meme's "textual content"; and *(iii)* identifying persuasion techniques across both the textual and visual components of the meme (a multimodal task). Furthermore, due to the complex nature of persuasion techniques, we present a hierarchy that groups the 22 persuasion techniques into several levels of categories. This became one of the attractive shared tasks in SemEval 2024, with 153 teams registered, 48 teams submitting results, and finally, 32 system description papers submitted.

1 Introduction

The rise of online social media platforms has enabled people to share their views and feelings openly. This increase in freedom of speech has significantly expanded the volume of digital content, offering valuable resources for initiatives like citizen journalism, raising public awareness, and supporting political campaigns. However, this freedom has also facilitated negative uses, leading to an increase in online hostility, as evidenced by the spread of content such as disinformation, hate speech, propaganda, and cyberbullying (Brooke, 2019; Joksimovic et al., 2019; Schmidt and Wiegand, 2017; Davidson et al., 2017; Da San Martino et al., 2019a; Van Hee et al., 2015).

Social media posts often combine various modalities, such as text, images, and videos. In recent years, *Internet memes* have become a prevalent form of content on these platforms. A meme is defined as "a collection of digital items that share common characteristics in content, form, or stance, which are created through association and widely circulated, imitated, or transformed over the Internet by numerous users." (Shifman, 2013) Memes generally consist of one or more images accompanied by textual content (Shifman, 2013; Suryawanshi et al., 2020). While memes are primarily aimed at humor, they can also convey persuasive narratives or content that may mislead audiences. To automatically identify such content, there have been research efforts directed towards addressing offensive content (Gandhi et al., 2020), identifying hate speech across different modalities (Gomez et al., 2020; Wu and Bhandary, 2020), and detecting propaganda techniques in memes (Dimitrov et al., 2021a).

Focusing on propaganda detection, research efforts have been specifically directed towards defining techniques and addressing the issue in news articles (Da San Martino et al., 2019), tweets (Alam et al., 2022b), memes (Dimitrov et al., 2021a), and textual content in multiple languages (Piskorski et al., 2023b). The associated shared tasks include SemEval-2020 Task 11 on news articles (Da San Martino et al., 2020), SemEval-2021 Task 6 on memes (Dimitrov et al., 2021b), WANLP-2022 and ArabicNLP-2023 focusing on Arabic

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 2009–2026

(Alam et al., 2022b; Hasanain et al., 2023), and SemEval-23 Task 3 on news articles in multiple languages (Piskorski et al., 2023b).

The SemEval-2024 shared task extends previous tasks but introduces multilinguality, covering four languages, and features the largest dataset in English, along with a new hierarchical evaluation method. It has attracted significant participation. The task consists of three subtasks and was run in two phases: *(i)* the development phase and *(ii)* the evaluation phase. In the remainder of this paper, we define the tasks, describe the datasets, and provide an overview of participating systems and their official scores.

2 Related Work

2.1 Persuasion Techniques Detection

Past research on propaganda detection focused on analyzing documents as a whole to assess whether they contained propaganda. Barrón-Cedeno et al. (2019) created a corpus categorized into *propaganda* and *non-propaganda*, exploring the writing style and readability levels. Their results indicated that using distant supervision combined with comprehensive representations could lead the model to predict the source of the article instead of accurately differentiating between propaganda and non-propaganda content. An alternative approach to research has concentrated on identifying the use of specific propaganda techniques within texts. For example, Habernal et al. (2017, 2018) constructed a corpus containing 1.3k arguments, each annotated with different fallacies directly associated with propaganda techniques.

Building on previous work, Da San Martino et al. (2019b) created a corpus of news articles annotated for eighteen fine-grained propaganda techniques, approaching the problem as a task of span detection and classification. The majority of these studies have primarily focused on English. To address this gap in multilingual settings, Piskorski et al. (2023c) developed a dataset of news articles encompassing nine languages (Piskorski et al., 2023c). This dataset has enabled research into developing multilingual models.

Focusing on multimodality, specifically on memes, Dimitrov et al. (2021a) developed a corpus consisting of 950 memes and investigated various transformer models for automatic detection.

2.2 Multimodal Content

Multimodal content has been effectively utilized for propagating information and generating positive impacts. At the same time, it has also been used to cause harm (Sharma et al., 2022) or spread mis- and dis-information (Alam et al., 2022a). Research in this area include predicting misleading information (Volkova et al., 2019), detecting deception (Glenski et al., 2019), emotions and propaganda (Abd Kadir et al., 2016), hateful memes (Kiela et al., 2020), and propaganda in images (Seo, 2014).

To address the problem, current state-of-the-art research includes fine-tuning transformer models such as ViLBERT (Lu et al., 2019), Multimodal Bitransformers (Kiela et al., 2019), and VisualBERT (Li et al., 2019). Several studies have also explored the use of prompting strategies for hateful meme classification (Cao et al., 2022), aiming for detection from both text and visual modalities by leveraging (Prakash et al., 2023). For a recent survey, please refer to the work by Hee et al. (2024), which reports on the role of multimodality and LLMs in hateful content moderation.

2.3 Related Shared Tasks

To foster community engagement, several shared tasks on propaganda detection have been organized in the past. *SemEval-2020 task 11 on Detection of Persuasion Techniques in News Articles* (Da San Martino et al., 2020) focused on news articles, and asked to detect the text spans where propaganda techniques are used, and to predict their type (14 techniques). Closely related to that is the *NLP4IF-2019 task on Fine-Grained Propaganda Detection* (Da San Martino et al., 2019), which asked to detect the spans of use in news articles of each of 18 propaganda techniques. The *SemEval-2023 task 3 Detecting the Category, the Framing, and the Persuasion Techniques in Online News in a Multi-lingual Setup* was focused on news articles covering nine languages Piskorski et al. (2023b). The WANLP'2022 and ArabicNLP'2023 shared task asked to detect the use of 20 propaganda techniques in Arabic tweets and news articles (Alam et al., 2022b; Hasanain et al., 2023).

The *SemEval-2021 Task 6 on Detection of Persuasion Techniques in Texts and Images* focused on identifying 22 persuasion techniques in memes (Dimitrov et al., 2021b). Following this prior work, we have significantly extended the size of the English dataset to 10K memes and added three sur-

prise languages. The task is divided into three subtasks and also presents the persuasion techniques in a newly formed hierarchy allowing for better system predictions in cases of low confidence when predicting persuasion.

3 Tasks and Dataset

3.1 Tasks

The objective of the shared task is to develop models capable of identifying persuasion techniques (see Table 2 for a list and Dimitrov et al. (2021b) for a detailed description). This involves one subtask focused solely on analyzing the textual content of memes and another two subtasks dedicated to a multimodal analysis, where both textual and visual content are examined together. The subtasks are defined as follows:

Subtask 1 (ST 1): Given only the "textual content" of a meme, identify which persuasion techniques, organized in a hierarchy, it uses. If the ancestor node of a technique is selected, only a partial reward is given. This is a multilingual hierarchical multilabel classification problem.

Subtask 2a (ST 2a): Given a meme, identify which persuasion techniques, organized in a hierarchy, are used both in the textual and in the visual content of the meme (multimodal task). If the ancestor node of a technique is selected, only a partial reward is given. This is a multilingual hierarchical multilabel classification problem.

Subtask 2b (ST 2b): Given a meme, identify whether it contains a persuasion technique. This is a binary classification problem.

	EN					BG	MK	AR
Subtask	Train	Val	Dev	Test	**Total**	Test	Test	Test
ST 1	7,000	500	1,000	1,500	10,000	436	259	100
ST 2a	7,000	500	1,000	1,500	10,000	436	259	120
ST 2b	1,200	150	300	600	2,250	100	100	160

Table 1: Number of memes for every language on each subtask and associated data splits. Note that **only** the test split contains all four languages. EN=English, BG=Bulgarian, MK=North Macedonian, AR=Arabic

3.2 Dataset

Collection: We collected English, Bulgarian, North Macedonian, and Arabic memes from our personal Facebook accounts by scraping public Facebook groups, which focus on politics, vaccines, COVID-19, gender equality, and the Russo-Ukrainian War. However, Facebook groups did not provide enough memes for North Macedonian and Arabic therefore we collected some of the memes for these languages from Instagram. We considered a meme to be a "*photograph style image with a short text on top of it*", and we removed examples that did not fit this definition, e.g., cartoon-style memes, memes whose textual content was strongly dominant or non-existent, memes with a single-color background image, etc.

Annotation: The list of persuasion techniques and the annotation process are as described in (Dimitrov et al., 2021b). For each meme, we first annotated its textual content, and then the entire meme. We performed each of these two annotations in two phases: in the first phase, the annotators independently annotated the memes; afterward, all annotators met together with a consolidator to discuss and select the final gold label(s). This process was applied to each language, however, for English we had an additional step in the process where an expert linguist reviewed random samples of consolidated memes and communicated his observations back to the team of annotators. This was done to ensure we maintained high-quality annotations throughout the whole annotation campaign, considering the high cognitive complexity of the task.

Statistics: Table 1 shows the number of memes for each subtask in all four languages. The data for every subtask was split into train, validation, dev, and test as shown in the table. We introduced a validation set to allow parameter optimization on a predefined set of data, making it comparable across different systems. Bulgarian, North Macedonian, and Arabic were only used for the test set as they were surprise languages.

Table 2 and Table 3 show the label distribution for all subtasks. *Transfer* and *Appeal to (Strong) Emotions* do not apply to text, i.e., to Subtask 1. For Subtasks 1 and 2a, each technique can be present at most once per example. From the persuasion technique distribution we can see that the dataset is extremely imbalanced with some labels being present in more than 50% of the memes (Smears) and others in less than 1% (Obfuscation, Intentional Vagueness, Confusion). Moreover, Figure 2 (in Appendix A) shows that most of the memes contain more than one persuasion technique.

Persuasion Techniques	Subtask 1 EN	BG	MK	AR	Subtask 2a EN	BG	MK	AR
Smears	2,838	41	25	17	5,159	320	220	63
Loaded Language	2,636	160	110	24	2,644	162	111	35
Name Calling/Labeling	2,284	140	83	26	2,294	148	95	33
Appeal to Authority	1,251	18	4	1	1,315	26	10	1
Black-and-White/Dictatorship	1,079	5	–	–	1,115	7	–	–
Slogans	994	62	23	–	1,024	68	26	–
Flag-Waving	834	28	6	1	1,179	43	14	2
Thought-Terminating Cliché	760	20	6	1	762	22	6	1
Glittering Generalities (Virtue)	703	5	–	2	991	29	5	2
Exaggeration/Minimisation	537	31	18	18	590	51	48	31
Appeal to Fear/Prejudice	527	35	13	8	643	73	52	43
Doubt	487	17	9	5	567	25	14	15
Repetition	442	19	3	1	445	19	3	1
Whataboutism	407	23	9	1	474	37	15	1
Causal Oversimplification	391	7	4	2	419	17	4	2
Bandwagon	144	2	–	1	157	6	–	1
Reductio ad Hitlerum	94	–	–	–	170	–	2	–
Straw Man	91	7	3	1	106	16	15	2
Presenting Irrelevant Data	87	3	1	1	91	–	1	3
Confusion	43	–	–	2	84	3	1	2
Transfer	–	–	–	–	2,286	141	113	–
Appeal to (Strong) Emotions	–	–	–	–	537	24	19	–
Total	**16,629**	**737**	**401**	**130**	**23,052**	**1,254**	**778**	**245**

Table 2: Persuasion techniques distribution for subtasks 1 and 2a in every language. For each technique, we show the number of instances.

Label	EN	BG	MK	AR
propagandistic	1,500	80	90	113
non propagandistic	750	20	10	47
Total	2,250	100	100	160

Table 3: Subtask 2b label distribution

We also observe a higher number of memes with 2 or more labels in ST2a which shows that a lot of memes require not only the text but the visual content to form enough context.

4 Evaluation Framework

4.1 Hierarchy

We introduce a hierarchy to allow the assignment of high-level categories in case of high uncertainty when predicting persuasion techniques. The persuasion techniques are grouped in a hierarchy, to be more precise a directed acyclic graph, as shown in Figure 3. The leaves of the hierarchy are the 22 persuasion techniques. The internal nodes are defined according to (Sourati et al., 2023; Piskorski et al., 2023a). Starting from the ROOT, we have the first level with *Ethos*, *Pathos*, and *Logos*. On the next level under Ethos – *Ad Hominem* and under Logos – *Justification* and *Reasoning*. Finally, Reasoning branches into *Distraction* and *Simplification*.

4.2 Evaluation Measures

Considering the hierarchical setup of the task, the evaluation metrics have to take into account the possibility of label assignment different than the original 22 persuasion techniques. Additionally, the metrics need to support a multilabel setting. We use adjusted F_1, $precision$, and $recall$ for hierarchical evaluation (Kiritchenko et al., 2006). For example, given the hierarchy in Figure 1, Let G be the ground truth value and H the predicted value, then to calculate the hierarchical measures we extend G to a set of its ancestor classes $S_{gold} = \{G, E, B, C\}$ and then do the same for H – $S_{pred} = \{H, E, B, C\}$. Then hierarchical $precision$, $recall$, and F_1 (hP, hR, and hF_1) would be:

$$hP = \frac{|S_{gold} \cap S_{pred}|}{|S_{pred}|} = \frac{|\{E, B, C\}|}{|\{H, E, B, C\}|} = \frac{3}{4} \quad (1)$$

$$hR = \frac{|S_{gold} \cap S_{pred}|}{|S_{gold}|} = \frac{|\{E, B, C\}|}{|\{G, E, B, C\}|} = \frac{3}{4} \quad (2)$$

$$hF_1 = \frac{2 \cdot hP \cdot hR}{hP + hR} = \frac{2 \cdot \frac{3}{4} \cdot \frac{3}{4}}{\frac{3}{4} + \frac{3}{4}} = \frac{3}{4} \quad (3)$$

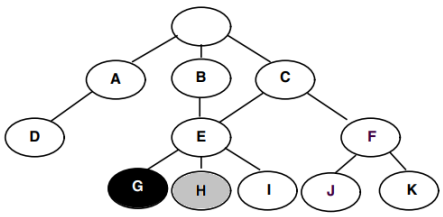


Figure 1: Example graph for hierarchical evaluation

Subtask	#team	#subm	#team	#subm
	EN Dev	EN Dev	EN Test	EN Test
ST 1	38	1159	35	130
ST 2a	11	61	14	28
ST 2b	20	457	20	51
Total	42	1677	42	209
	BG Test	BG Test	MK Test	MK Test
ST 1	20	29	20	29
ST 2a	8	13	8	10
ST 2b	15	20	15	21
Total	27	62	27	60
	AR Test	AR Test		
ST 1	17	36		
ST 2a	8	19		
ST 2b	15	24		
Total	24	79		

Table 4: Submission statistics. Note that only English has dev submissions, as the other languages were only released for test. *#team*: Number of teams that submitted results; *#subm*: Number of submissions.

Subtasks 1 and 2a are hierarchical multilabel classification problems. We used $hierarchical - F_1$ as the official evaluation measure. We also computed $hierarchical\ precision$ and $hierarchical\ recall$.

Subtask 2b is a binary classification problem. We used $macro\ F_1$ as the official evaluation measure. We also computed $micro\ F_1$.

4.3 Task Organization

The shared task was run in two phases:

Development Phase: During the development phase, we made *training* and *development* sets available for the participants. However, gold standard labels were not released for the development set. The participants submittd systems' results on the development set. They could make an unlimited number of submissions, and the best score for each team, regardless of the submission time, was shown in real time on a public leaderboard.

Test Phase: In this phase, we have released the test set and the *development* set together with the gold labels. The participants were given a week to submit their final predictions on the *test* set. It is important to note that the test data included memes in three additional languages such as Bulgarian, North Macedonian, and Arabic, which were not disclosed to the participants in advance as surprise languages. Similar to the development phase, participants could submit multiple entries; however, they have not received any feedback on their performance. Only the latest submission from each team was considered official and used to determine the final team rankings. Overall, 153 teams registered for the task, out of which 48 made official submissions. Moreover, 24 teams submitted results for all four languages. Specifically, 17 teams submitted results for all languages for ST1, 8 for ST2a, and 14 for ST2b, respectively. The total number of submissions across both phases was 2,087, with 1,677 on the development set and 410 on the test set. More details on submission statistics can be found in Table 4.

The results for the development and the test phases are available on the leaderboard page.[1] After the competition was over, we left the submission system open for the test dataset for post-shared task evaluations and to monitor the state of the art for the different subtasks across the languages.

4.4 Baseline Systems

Due to the highly imbalanced dataset, as seen in Table 2, the baseline for each subtask is the most common label or majority class baseline, i.e., for each meme, we make a prediction with the most frequent label. *Smears* is the most frequent label for Subtasks 1 and 2a, and *propagandistic* is the most frequent label for Subtask 2b. Note that the baseline is chosen according to the most common label across all languages.

5 Results

5.1 English Subtasks

The results for the three English subtasks are presented in Tables 5 and 6. All systems outperformed the baseline and the winning system is noticeably better than the second in subtasks 1 and 2a. In Subtask 2b there are three teams with top performance, two winning systems ex-aequo, and a third with a 0.001 difference in F_1.

We now briefly describe some of the top systems for each subtask. In Subtask 1 **914isthebest** (Li et al., 2024a) developed a transformer-based model

[1] https://propaganda.math.unipd.it/semeval2024task4/leaderboard.php

with in-domain pre-training. For system training, the training dataset was augmented following a Chain-of-Thought-based data augmentation approach using GPT-3.5. The main classification architecture includes four RoBERTa models and one DeBERTa model initialized using different random seeds. A soft voting approach, which averages the predicted probabilities of each label from all five models, is used to predict labels.

In Subtask 2a **HierarchyEverywhere** (Ghahroodi and Asgari, 2024) adapted the hierarchical text classification (HTC) model to the task by placing "propagandistic" and "non-propagandistic" nodes at the initial level and utilizing the "[CLS]" Token between sentences in memes enhanced model performance (Wang et al., 2022). Moreover, they employed additional datasets. Interestingly, the image component of memes was disregarded, and only the textual content was provided to the model. Furthermore, for all the sub-tasks that are non-English, Google Translation API was used to translate them into English.

In Subtask 2b, **LMEME** (Li et al., 2024b) proposed a detection system that employs a Teacher Student Fusion framework. Initially, a Large Language Model serves as the teacher, engaging in abductive reasoning on multimodal inputs to generate background knowledge on persuasion techniques, assisting in the training of a smaller downstream model. The student model adopts CLIP as an encoder for text and image features, incorporating an attention mechanism for modality alignment.

5.2 Bulgarian Subtasks

The results for the Bulgarian subtasks are reported in Tables 7 and 8. For Subtask 1, seventeen out of nineteen systems outperformed the baseline; for Subtask 2a, four out of seven systems outperformed the baseline; and for Subtask 2b, all systems outperformed the baseline.

We briefly describe some of the top systems for each subtask. The top system, **OtterlyObsessedWithSemantics** (Wunderle et al., 2024) for Subtask 1, used a custom classification head that is designed to be applied atop a large language model. For the non-English test sets, the system was used after translating all documents to English using GPT-4.

For Subtask 2a, the top system is **BCAmirs** (Abaskohi et al., 2024). It involved using GPT-4 to generate a descriptive caption of the meme. The caption is then combined

R	Team	hF1	hP	hR
	English - Subtask 1			
1	914isthebest	0.752	0.684	0.836
2	BCAmirs	0.699	0.668	0.732
3	OtterlyObsessedWithSemantics	0.697	0.648	0.755
4	TUMnlp	0.674	0.638	0.714
5	GreyBox	0.670	0.652	0.688
6	NLPNCHU	0.663	0.610	0.726
7	Puer	0.660	0.648	0.673
8	EURECOM	0.655	0.628	0.685
9	SuteAlbastre	0.652	0.633	0.673
10	UMUTeam	0.648	0.708	0.597
11	RDproj	0.643	0.575	0.728
12	HierarchyEverywhere	0.643	0.636	0.649
13	nowhash	0.641	0.612	0.673
14	ShefCDTeam	0.640	0.662	0.618
15	Pauk	0.627	0.716	0.557
16	IUSTNLPLAB	0.625	0.632	0.618
17	whatdoyoumeme	0.617	0.598	0.638
18	LomonosovMSU	0.613	0.712	0.539
19	SoftMiner	0.607	0.649	0.569
20	MagnumJUCSE	0.603	0.547	0.673
21	IITK	0.591	0.596	0.586
22	CLaC	0.578	0.501	0.685
23	BAMBAS	0.577	0.501	0.679
24	MemeSifters	0.575	0.576	0.573
25	fralak	0.557	0.478	0.668
26	IIITG	0.526	0.614	0.459
27	Two	0.522	0.526	0.518
28	Scalar	0.505	0.433	0.606
29	SINAI	0.425	0.312	0.667
30	McRock	0.423	0.301	0.708
31	Baseline	0.369	0.477	0.300
32	WhatsaMeme	0.347	0.347	0.346
33	IIMAS1UTM1LaSalle	0.199	0.755	0.115
	English - Subtask 2a			
1	HierarchyEverywhere	0.746	0.867	0.655
2	NLPNCHU	0.707	0.782	0.645
3	BCAmirs	0.705	0.784	0.641
4	UMUTeam	0.690	0.768	0.627
5	SuteAlbastre	0.685	0.718	0.655
6	TUMnlp	0.677	0.781	0.598
7	Pauk	0.675	0.745	0.617
8	CodeMeme	0.666	0.607	0.739
9	LomonosovMSU	0.656	0.792	0.560
10	IITK	0.636	0.763	0.545
11	BERTastic	0.613	0.816	0.491
12	BDA	0.504	0.515	0.493
13	Baseline	0.447	0.688	0.331
14	WhatsaMeme	0.366	0.313	0.440

Table 5: Official results for English - Subtasks 1 and 2a. Runs ranked by the official measure (Hierarchical F1).

Rank	Team	F1 macro	F1 micro
1	LMEME	0.810	0.825
2	SuteAlbastre	0.810	0.835
3	DUTIR938	0.809	0.837
4	BCAmirs	0.803	0.825
5	Snarci	0.799	0.827
6	BDA	0.793	0.823
7	NLPNCHU	0.788	0.822
8	UMUTeam	0.787	0.807
9	TUMnlp	0.784	0.802
10	CodeMeme	0.782	0.807
11	LomonosovMSU	0.772	0.798
12	BERTastic	0.716	0.762
13	Hidetsune	0.714	0.790
14	Scalar	0.702	0.753
15	SheffieldVeraAI	0.642	0.687
16	HierarchyEverywhere	0.563	0.662
17	WhatsaMeme	0.515	0.530
18	nowhash	0.498	0.515
19	IITK	0.483	0.490
20	Baseline	0.250	0.333

Table 6: Official results for English - Subtask 2b. Runs ranked by the official measure (Hierarchical F1).

with the meme text, before being passed to a RoBERTa model. A vision encoder utilizing a pre-trained vision transformer model (CLIP-ViT), is used to encode and analyze the meme image. Finally, a multi-layer perceptron classifier takes the combined visual and textual representations and classifies the meme. The models used were monolingual, and thus, for non-English tasks, the system was applied to test sets translated using Google Translate.

In Subtask 2b, the top system is **LMEME** (Li et al., 2024b), which was also the top system for Subtask 2b in English, and presented in Section 5.1.

5.3 North Macedonian Subtasks

The results for the North Macedonian subtasks are presented in Tables 9 and 10. Our observations for the North Macedonian subtasks closely align with those for the Bulgarian subtasks. For subtasks 1 and 2a, a few teams were unable to surpass the baseline results. However, for Subtask 2b, all teams exceeded the baseline performance.

As with the Bulgarian Subtask 1 and Subtask 2a, the top systems for North Macedonian were also **OtterlyObsessedWithSemantics** (Wunderle et al., 2024) and **BCAmirs**, respectively.

Team **BERTastic** (Mahmoud and Nakov, 2024) achieved the best performance for Subtask 2b. The system uses three representations of the input meme, including the image, associated text, and a generic description of the meme generated

R	Team	hF1	hP	hR
Bulgarian - Subtask 1				
1	OtterlyObsessedWithSemantics	0.568	0.520	0.627
2	RDproj	0.541	0.435	0.714
3	NLPNCHU	0.517	0.536	0.500
4	MagnumJUCSE	0.500	0.470	0.533
5	nowhash	0.486	0.460	0.516
6	MemeSifters	0.481	0.491	0.472
7	GreyBox	0.476	0.438	0.522
8	whatdoyoumeme	0.473	0.502	0.446
9	HierarchyEverywhere	0.468	0.483	0.453
10	fralak	0.464	0.374	0.613
11	914isthebest	0.463	0.477	0.450
12	CLaC	0.449	0.400	0.512
13	BCAmirs	0.448	0.387	0.533
14	IITK	0.434	0.404	0.470
15	ShefCDTeam	0.366	0.454	0.307
16	EURECOM	0.345	0.367	0.325
17	SINAI	0.341	0.214	0.849
18	Baseline	0.284	0.319	0.256
19	SuteAlbastre	0.236	0.134	1.000
20	IIMAS1UTM1LaSalle	0.183	0.654	0.107
Bulgarian - Subtask 2a				
1	BCAmirs	0.627	0.703	0.566
2	SuteAlbastre	0.611	0.660	0.569
3	NLPNCHU	0.549	0.707	0.448
4	BERTastic	0.544	0.812	0.409
5	Baseline	0.500	0.804	0.363
6	BDA	0.483	0.523	0.450
7	HierarchyEverywhere	0.464	0.671	0.355
8	IITK	0.446	0.541	0.379

Table 7: Bulgarian - Subtasks 1 and 2a

Bulgarian - Subtask 2b			
Rank	**Team**	**F1 macro**	**F1 micro**
1	LMEME	0.671	0.810
2	Snarci	0.668	0.840
3	BERTastic	0.662	0.750
4	BCAmirs	0.647	0.770
5	NLPNCHU	0.647	0.820
6	MemeSifters	0.611	0.830
7	SuteAlbastre	0.594	0.650
8	SheffieldVeraAI	0.536	0.570
9	BDA	0.506	0.620
10	HierarchyEverywhere	0.485	0.630
11	IITK	0.473	0.530
12	DUTIR938	0.434	0.570
13	nowhash	0.434	0.450
14	Hidetsune	0.327	0.330
15	Baseline	0.167	0.200

Table 8: Bulgarian - Subtask 2b

R	Team	hF1	hP	hR
North Macedonian - Subtask 1				
1	OtterlyObsessedWithSemantics	0.512	0.518	0.507
2	RDproj	0.499	0.434	0.587
3	MagnumJUCSE	0.483	0.486	0.480
4	fralak	0.464	0.359	0.658
5	NLPNCHU	0.462	0.546	0.400
6	EURECOM	0.442	0.520	0.384
7	MemeSifters	0.441	0.539	0.373
8	GreyBox	0.434	0.440	0.429
9	nowhash	0.426	0.414	0.438
10	HierarchyEverywhere	0.417	0.486	0.365
11	CLaC	0.395	0.371	0.422
12	BCAmirs	0.393	0.332	0.482
13	IITK	0.383	0.344	0.432
14	914isthebest	0.369	0.401	0.341
15	whatdoyoumeme	0.362	0.399	0.331
16	ShefCDTeam	0.319	0.436	0.251
17	Baseline	0.307	0.314	0.300
18	SINAI	0.301	0.183	0.846
19	SuteAlbastre	0.204	0.113	0.996
20	IIMAS1UTM1LaSalle	0.137	0.529	0.079
North Macedonian - Subtask 2a				
1	BCAmirs	0.637	0.750	0.553
2	SuteAlbastre	0.576	0.492	0.692
3	BERTastic	0.573	0.866	0.428
4	Baseline	0.555	0.902	0.401
5	BDA	0.501	0.546	0.463
6	NLPNCHU	0.487	0.706	0.372
7	IITK	0.440	0.545	0.369
8	HierarchyEverywhere	0.357	0.689	0.241

Table 9: North Macedonian - Subtasks 1 and 2a

by a vision-language model. A multilingual model, MPNet, was used to extract embeddings from text elements, while a multimodal multilingual model, CLIP-ViT-B-32, was used to represent both text and image. All extracted features were fused into a single feature vector, followed by logistic regression for classification.

5.4 Arabic Subtasks

In Tables 11 and 12, we report the results for Arabic subtasks. Here, we also observe similar patterns to Bulgarian and North Macedonian. The top systems for Subtask 1 and Subtask 2a are also **OtterlyObsessedWithSemantics** (Wunderle et al., 2024) and **BCAmirs**, respectively. **BCAmirs** also achieves the top performance for Subtask 2b.

Considering all non-English languages, we see that many systems struggled to surpass the baseline for Subtask 2a specifically. This can be due to the difficult nature of this subtask, as it is a hierarchical multilabel classification task that also requires considering multimodal content. Such difficulty also affected the number of participants, with a relatively smaller number of systems submitted to this

North Macedonian - Subtask 2b			
Rank	**Team**	**F1 macro**	**F1 micro**
1	BERTastic	0.686	0.840
2	MemeSifters	0.660	0.900
3	LMEME	0.591	0.780
4	BCAmirs	0.561	0.770
5	NLPNCHU	0.520	0.790
6	HierarchyEverywhere	0.506	0.620
7	IITK	0.485	0.630
8	Snarci	0.479	0.720
9	DUTIR938	0.469	0.660
10	SheffieldVeraAI	0.458	0.510
11	BDA	0.435	0.600
12	nowhash	0.429	0.520
13	Hidetsune	0.389	0.460
14	SuteAlbastre	0.177	0.180
15	Baseline	0.091	0.100

Table 10: North Macedonian - Subtask 2b

R	Team	hF1	hP	hR
Arabic - Subtask 1				
1	OtterlyObsessedWithSemantics	0.476	0.391	0.607
2	NLPNCHU	0.475	0.428	0.533
3	fralak	0.428	0.309	0.698
4	whatdoyoumeme	0.424	0.328	0.600
5	RDproj	0.411	0.333	0.537
6	IITK	0.408	0.339	0.512
7	HierarchyEverywhere	0.405	0.356	0.470
8	nowhash	0.404	0.360	0.460
9	BCAmirs	0.396	0.320	0.519
10	MagnumJUCSE	0.395	0.346	0.460
11	CLaC	0.381	0.308	0.498
12	MemeSifters	0.360	0.355	0.365
13	914isthebest	0.360	0.314	0.421
14	Baseline	0.359	0.350	0.368
15	SINAI	0.258	0.154	0.793
16	SuteAlbastre	0.234	0.198	0.288
17	EURECOM	0.177	0.343	0.119
Arabic - Subtask 2a				
1	BCAmirs	0.526	0.553	0.502
2	SuteAlbastre	0.516	0.469	0.573
3	Baseline	0.486	0.650	0.389
4	NLPNCHU	0.483	0.595	0.407
5	IITK	0.455	0.457	0.453
6	HierarchyEverywhere	0.437	0.510	0.382
7	BDA	0.416	0.382	0.457
8	BERTastic	0.388	0.613	0.284

Table 11: Arabic - Subtasks 1 and 2a

Arabic - Subtask 2b			
Rank	**Team**	**F1 macro**	**F1 micro**
1	BCAmirs	0.615	0.631
2	SheffieldVeraAI	0.610	0.613
3	BERTastic	0.603	0.606
4	NLPNCHU	0.585	0.594
5	HierarchyEverywhere	0.562	0.669
6	MemeSifters	0.557	0.694
7	Snarci	0.555	0.556
8	Hidetsune	0.528	0.544
9	BDA	0.510	0.606
10	SuteAlbastre	0.501	0.544
11	nowhash	0.498	0.531
12	DUTIR938	0.469	0.519
13	IITK	0.467	0.469
14	LMEME	0.362	0.388
15	Baseline	0.227	0.294

Table 12: Arabic - Subtask 2b

subtask compared to the other two subtasks.

6 Conclusions and Future Work

We presented SemEval-2024 Task 4 on *Multilingual Detection of Persuasion Techniques in Memes*. The task consists of detecting persuasion techniques in memes in a multimodal setting. The task offered a significantly larger dataset for English (10K memes) than previous ones, and three surprise languages: Arabic, Bulgarian, and North Macedonian.

The task attracted a lot of attention: 153 teams registered for the task and 30 teams submitted a task description paper. Fine-tuning transformer-based architectures was the most dominant approach followed by most teams. The majority of teams participating in Subtask 2 considered both the text and image components of the data, utilizing corresponding transformer models. Finally, several teams designed hierarchical classification techniques, to tackle the hierarchy of labels in Subtask 1 and Subtask 2a. As for the surprise languages, at least a third of the submitting teams used automatic translation to translate the datasets into English.

7 Limitations

The dataset we have collected originates from various public Facebook groups, with a primary focus on politics. Consequently, the representativeness of this dataset may be limited for other domains and topics. The highly imbalanced distribution of the labels in the dataset may affect the model's performance. Therefore, it is important to develop models with this aspect in mind.

Ethics and Broader Impact

Our dataset solely comprises memes, and we have not collected any user information; therefore, the privacy risk is nonexistent.

Any biases present in the dataset are unintentional, and our intention is not to cause harm to any group or individual. It's important to acknowledge that annotating propaganda techniques involves a degree of subjectivity, making biases in our gold-labeled data or label distribution unavoidable. To mitigate these concerns, we have collected examples from a diverse range of users and groups. Furthermore, we adhere to a well-defined schema with clear definitions, which has enabled us to achieve high inter-annotator agreement. Additionally, our annotation team was diverse, consisting of six members, including both females and males.

We advise researchers of the risk that our dataset could be exploited to biasly moderate memes, potentially due to biases related to demographics or specifics in the text. To prevent this, the implementation of human moderation is crucial.

References

Amirhossein Abaskohi, Amirhossein Dabiriaghdam, Lele Wang, and Giuseppe Carenini. 2024. Bcamirs at semeval-2024 task 4: Beyond words: A multimodal and multilingual exploration of persuasion in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1402–1412, Mexico City, Mexico. Association for Computational Linguistics.

Shamsiah Abd Kadir, Anitawati Lokman, and T. Tsuchiya. 2016. Emotion and techniques of propaganda in YouTube videos. *Indian Journal of Science and Technology*, Vol (9).

Firoj Alam, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2022a. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6625–6643, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Firoj Alam, Hamdy Mubarak, Wajdi Zaghouani, Giovanni Da San Martino, and Preslav Nakov. 2022b. Overview of the WANLP 2022 shared task on propaganda detection in Arabic. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop*, Abu Dhabi, UAE.

Ion Anghelina, Gabriel Buță, and Alexandru Enache. 2024. Sutealbastre at semeval-2024 task 4: Predicting propaganda techniques in multilingual memes

using joint text and vision transformers. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 430–436, Mexico City, Mexico. Association for Computational Linguistics.

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Sian Brooke. 2019. "condescending, rude, assholes": Framing gender and hostility on stack overflow. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 172–180, Florence, Italy. Association for Computational Linguistics.

Rui Cao, Roy Ka-Wei Lee, Wen-Haw Chong, and Jing Jiang. 2022. Prompting for multimodal hateful meme classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 321–332, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nishan Chatterjee, Marko Pranjic, Boshko Koloski, Lidia Pivovarova, and Senja Pollak. 2024. whatdoyoumeme at semeval-2024 task 4: Hierarchical-label aware cross-lingual persuasion detection using translated texts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1548–1554, Mexico City, Mexico. Association for Computational Linguistics.

Shreenaga Chikoti, Shrey Mehta, and Ashutosh Modi. 2024. Iitk at semeval-2024 task 4: Hierarchical embeddings for detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1790–1798, Mexico City, Mexico. Association for Computational Linguistics.

Abu Nowhash Chowdhury and Michal Ptaszynski. 2024. nowhash at semeval-2024 task 4: Exploiting fusion of transformers for detecting persuasion techniques in multilingual memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 133–138, Mexico City, Mexico. Association for Computational Linguistics.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '20, Barcelona, Spain.

Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019. Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, NLP4IF '19, pages 162–170, Hong Kong, China.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019a. Fine-grained analysis of propaganda in news articles. In *EMNLP*.

Giovanni Da San Martino, Seunghak Yu, Alberto Barron-Cedeno, Rostislav Petrov, and Preslav Nakov. 2019b. Fine-grained analysis of propaganda in news articles. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, EMNLP '19, pages 5636–5646, Hong Kong, China.

Jiaxu Dao, Zhuoying Li, Youbang Su, and Wensheng Gong. 2024. Puer at semeval-2024 task 4: Fine-tuning pre-trained language models for meme persuasion technique detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 64–69, Mexico City, Mexico. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11 of *AAAI '17*.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021a. Detecting propaganda techniques in memes. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '21, pages 6603–6617.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021b. Task 6 at SemEval-2021: Detection of persuasion techniques in texts and images. In *Proceedings of the 15th International Workshop on Semantic Evaluation*, SemEval '21, Bangkok, Thailand.

Shreyansh Gandhi, Samrat Kokkula, Abon Chaudhuri, Alessandro Magnani, Theban Stanley, Behzad Ahmadi, Venkatesh Kandaswamy, Omer Ovenc, and Shie Mannor. 2020. Scalable detection of offensive and non-compliant content / logo in product images. *WACV*, pages 2236–2245.

Omid Ghahroodi and Ehsaneddin Asgari. 2024. Hierarchyeverywhere at semeval-2024 task 4: Detection of persuasion techniques in memes using hierarchical text classifier. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1738–1743, Mexico City, Mexico. Association for Computational Linguistics.

Meredith Gibbons, Maggie Mi, Xingyi Song, and Aline Villavicencio. 2024. Shefcdteam at semeval-2024 task 4: A text-to-text model for multi-label classification. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1872–1879, Mexico City, Mexico. Association for Computational Linguistics.

Maria Glenski, E. Ayton, J. Mendoza, and Svitlana Volkova. 2019. Multilingual multimodal digital deception detection and disinformation spread across social platforms. *ArXiv*, abs/1909.05838.

Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2020. Exploring hate speech detection in multimodal publications. In *WACV*, pages 1470–1478.

Charlie Grimshaw, Kalina Bontcheva, and Xingyi Song. 2024. Sheffieldveraai at semeval-2024 task 4: Prompting and fine-tuning a large vision-language model for binary classification of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2035–2040, Mexico City, Mexico. Association for Computational Linguistics.

Shih-Wei Guo and Yao-Chung Fan. 2024. Nlpnchu at semeval-2024 task 4: A comparison of mdhc strategy and in-domain pre-training for multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1880–1887, Mexico City, Mexico. Association for Computational Linguistics.

Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational argumentation meets serious games. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, EMNLP '17, pages 7–12, Copenhagen, Denmark.

Ivan Habernal, Patrick Pauli, and Iryna Gurevych. 2018. Adapting serious game for fallacious argumentation to German: Pitfalls, insights, and best practices. In *LREC*. European Language Resources Association (ELRA).

Maram Hasanain, Firoj Alam, Hamdy Mubarak, Samir Abdaljalil, Wajdi Zaghouani, Preslav Nakov, Giovanni Da San Martino, and Abed Freihat. 2023. ArAIEval shared task: Persuasion techniques and disinformation detection in Arabic text. In *Proceedings of ArabicNLP 2023*, pages 483–493, Singapore (Hybrid). Association for Computational Linguistics.

Ming Shan Hee, Shivam Sharma, Rui Cao, Palash Nandi, Preslav Nakov, Tanmoy Chakraborty, and Roy Ka-Wei Lee. 2024. Recent advances in hate speech moderation: Multimodality and the role of large models. *arXiv preprint arXiv:2401.16727*.

Srecko Joksimovic, Ryan S. Baker, Jaclyn Ocumpaugh, Juan Miguel L. Andres, Ivan Tot, Elle Yuan Wang, and Shane Dawson. 2019. Automated identification of verbally abusive behaviors in online discussions. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 36–45, Florence, Italy. Association for Computational Linguistics.

Adnan Khurshid and Dipankar Das. 2024. Magnum jucse at semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1004–1007, Mexico City, Mexico. Association for Computational Linguistics.

Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, and Davide Testuggine. 2019. Supervised multimodal bitransformers for classifying images and text. In *Proceedings of the NeurIPS 2019 Workshop on Visually Grounded Interaction and Language*, ViGIL@NeurIPS '19.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, NeurIPS '20.

Svetlana Kiritchenko, Richard Nock, and Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. volume 4013, pages 395–406.

Katarina Laken. 2024. Fralak at semeval-2024 task 4: combining rnn-generated hierarchy paths with simple neural nets for hierarchical multilabel text classification in a multilingual zero-shot setting. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 583–588, Mexico City, Mexico. Association for Computational Linguistics.

Dailin Li, Chuhan Wang, Xin Zou, Junlong Wang, Peng Chen, Jian Wang, Liang Yang, and Hongfei Lin. 2024a. 914isthebest at semeval-2024 task 4: Cot-based data augmentation strategy for persuasion techniques detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Shiyi Li, Yike Wang, Liang Yang, Shaowu Zhang, and Hongfei Lin. 2024b. Lmeme at semeval-2024 task 4: Teacher student fusion - integrating clip with llms for enhanced persuasion detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 615–620, Mexico City, Mexico. Association for Computational Linguistics.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the Conference on Neural Information Processing Systems*, NeurIPS '19, Vancouver, Canada.

Tarek Mahmoud and Preslav Nakov. 2024. Bertastic at semeval-2024 task 4: State-of-the-art multilingual propaganda detection in memes via zero-shot learning with vision-language models. In *Proceedings of*

the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 490–497, Mexico City, Mexico. Association for Computational Linguistics.

Kota Shamanth Ramanath Nayak and Leila Kosseim. 2024. Clac at semeval-2024 task 4: Decoding persuasion in memes – an ensemble of language models with paraphrase augmentation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 175–180, Mexico City, Mexico. Association for Computational Linguistics.

Mohammad Osoolian, Erfan Moosavi Monazzah, and Sauleh Eetemadi. 2024. Iustnlplab at semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1081–1085, Mexico City, Mexico. Association for Computational Linguistics.

ronghao pan, José Antonio García-Díaz, and Rafael Valencia-García. 2024. Umuteam at semeval-2024 task 4: Multimodal identification of persuasive techniques in memes through large language models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 642–652, Mexico City, Mexico. Association for Computational Linguistics.

Matt Pauk and Maria Leonor Pacheco. 2024. Pauk at semeval-2024 task 4: A neuro-symbolic method for consistent classification of propaganda techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1413–1423, Mexico City, Mexico. Association for Computational Linguistics.

Youri Peskine, Raphael Troncy, and Paolo Papotti. 2024. Eurecom at semeval-2024 task 4: Hierarchical loss and model ensembling in detecting persuasion techniques. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1166–1171, Mexico City, Mexico. Association for Computational Linguistics.

Jakub Piskorski, Nicolas Stefanovitch, Valerie-Anne Bausier, Nicolo Faggiani, Jens Linge, Sopho Kharazi, Nikolaos Nikolaidis, Giulia Teodori, Bertrand De Longueville, Brian Doherty, Jason Gonin, Camelia Ignat, Bonka Kotseva, Eleonora Mantica, Lorena Marcaletti, Enrico Rossi, Alessio Spadaro, Marco Verile, Giovanni Da San Martino, Firoj Alam, and Preslav Nakov. 2023a. News categorization, framing and persuasion techniques: Annotation guidelines. Technical Report JRC-132862, European Commission Joint Research Centre, Ispra (Italy).

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023b. SemEval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Jakub Piskorski, Nicolas Stefanovitch, Nikolaos Nikolaidis, Giovanni Da San Martino, and Preslav Nakov. 2023c. Multilingual multifaceted understanding of online news in terms of genre, framing, and persuasion techniques. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3001–3022, Toronto, Canada. Association for Computational Linguistics.

Nirmalendu Prakash, Han Wang, Nguyen Khoi Hoang, Ming Shan Hee, and Roy Ka-Wei Lee. 2023. PromptMTopic: Unsupervised multimodal topic modeling of memes using large language models. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 621–631, New York, NY, USA. Association for Computing Machinery.

Nathan Roll and Calbert Graham. 2024. Greybox at semeval-2024 task 4: Progressive fine-tuning (for multilingual detection of propaganda techniques). In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 875–880, Mexico City, Mexico. Association for Computational Linguistics.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Hyunjin Seo. 2014. Visual propaganda in the age of social media: An empirical analysis of Twitter images during the 2012 Israeli–Hamas conflict. *Visual Communication Quarterly*, 21(3).

Shivam Sharma, Firoj Alam, Md. Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty. 2022. Detecting and understanding harmful memes: A survey. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, IJCAI '22, pages 5597–5606, Vienna, Austria. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Victoria Sherratt, Sedat Dogan, Ifeoluwa Wuraola, Lydia Bryan-Smith, Oyinkansola Onwuchekwa, and Nina Dethlefs. 2024. Bda at semeval-2024 task 4: Detection of persuasion in memes across languages with ensemble learning and external knowledge. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 123–132, Mexico City, Mexico. Association for Computational Linguistics.

Limor Shifman. 2013. *Memes in digital culture*. MIT press.

Marco Siino. 2024. Mcrock at semeval-2024 task 4: Mistral 7b for multilingual detection of persuasion techniques in memes. In *Proceedings of the*

18th International Workshop on Semantic Evaluation (SemEval-2024), pages 53–59, Mexico City, Mexico. Association for Computational Linguistics.

Gleb Skiba, Mikhail Pukemo, Dmitry Melikhov, and Konstantin Vorontsov. 2024. Lomonosovmsu at semeval-2024 task 4: Comparing llms and embedder models to identifying propaganda techniques in the content of memes in english for subtasks №1, №2a, and №2b. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1555–1559, Mexico City, Mexico. Association for Computational Linguistics.

Zhivar Sourati, Vishnu Priya Prasanna Venkatesh, Darshan Deshpande, Himanshu Rawlani, Filip Ilievski, Hông-Ân Sandlin, and Alain Mermoud. 2023. Robust and explainable identification of logical fallacies in natural language arguments. *Know.-Based Syst.*, 266(C).

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *TRAC*, pages 32–41.

Hidetsune Takahashi. 2024. Hidetsune at semeval-2024 task 4: An application of machine learning to multilingual propagandistic memes identification using machine translation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 363–366, Mexico City, Mexico. Association for Computational Linguistics.

Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Veronique Hoste. 2015. Detection and fine-grained classification of cyberbullying events. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 672–680, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Arthur Vasconcelos, Luiz Felipe de Melo, Eduardo Goncalves, Eduardo Bezerra, Aline Paes, and Alexandre Plastino. 2024. Bambas at semeval-2024 task 4: How far can we get without looking at hierarchies? In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 442–449, Mexico City, Mexico. Association for Computational Linguistics.

Svitlana Volkova, Ellyn Ayton, Dustin L. Arendt, Zhuanyi Huang, and Brian Hutchinson. 2019. Explaining multimodal deceptive news prediction models. In *Proceedings of the International Conference on Web and Social Media*, ICWSM '19, Munich, Germany.

Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022. HPT: Hierarchy-aware prompt tuning for hierarchical text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3740–3751, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ching Seh Wu and Unnathi Bhandary. 2020. Detection of hate speech in videos using machine learning. In *CSCI*, pages 585–590.

Julia Wunderle, Julian Schubert, Antonella Cacciatore, Albin Zehe, Jan Pfister, and Andreas Hotho. 2024. Otterlyobsessedwithsemantics at semeval-2024 task 4: Developing a hierarchical multi-label classification head for large language models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 589–599, Mexico City, Mexico. Association for Computational Linguistics.

Erchen Yu, Junlong Wang, Xuening Qiao, Jiewei Qi, Zhaoqing Li, Hongfei Lin, Linlin Zong, and Bo Xu. 2024. Dutir938 at semeval-2024 task 4: Semi-supervised learning and model ensemble for persuasion techniques detection in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 629–635, Mexico City, Mexico. Association for Computational Linguistics.

Luca Zedda, Alessandra Perniciano, Andrea Loddo, Cecilia Di Ruberto, Manuela Sanguinetti, and Maurizio Atzori. 2024. Snarci at semeval-2024 task 4: Themis model for binary classification of memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 840–845, Mexico City, Mexico. Association for Computational Linguistics.

Yuhang Zhu. 2024. Rdproj at semeval-2024 task 4: An ensemble learning approach for multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 181–187, Mexico City, Mexico. Association for Computational Linguistics.

A Additional Dataset Details

Figure 2 shows statistics about the distribution of the number of persuasion techniques per meme for Subtasks 1 and 2a. The techniques hierarchy in 3 shows the details of coarse and fine-grained categories.

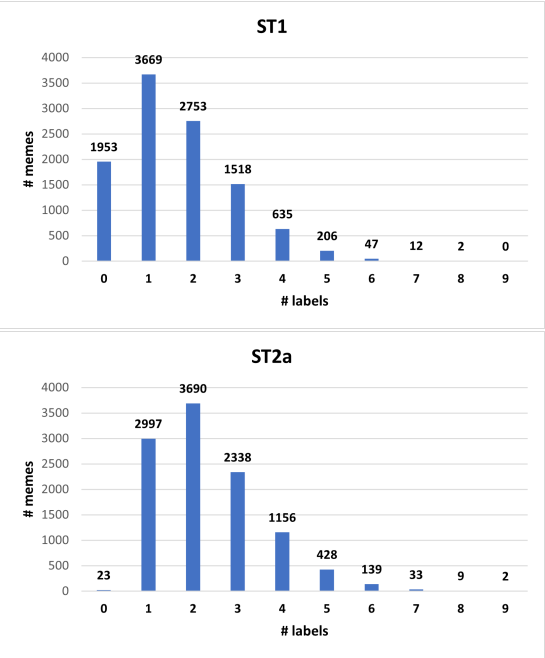


Figure 2: Subtasks 1 and 2a number of labels distributions.

B Overview of Participating Systems

In this section, we provide a summary of the approach followed by each of the participating systems.

BDA (Sherratt et al., 2024) The team participated in both Subtask 2a and Subtask 2b. For Subtask 2a, the proposed architecture is an ensemble of models operating on two modalities, text and images. For text, an ensemble of mBERT and XLM-RoBERTa is used, while CLIP and a monolingual BERT model is used to process visual entities extracted from images using Google Vision. Finally, a late fusion engine is used to merge predictions; generate additional translated task data; and modify the prediction confidence threshold based on the task hierarchy. As for Subtask 2b, the system is an ensemble of three models: 1) XLM-RoBERTa, that is trained on augmented task data, 2) VGG19 trained on task images and 3) a BERT model trained on visual entities extracted from the images using Google Vision. Late fusion is applied to join predictions from the models.

OtterlyObsessedWithSemantics (Wunderle et al., 2024) For Subtask 1, a custom classification head that is designed to be applied atop of a large language model was used. This approach includes reconstructing the hierarchy across multiple fully connected layers, allowing for incorporation of previous foundational decisions in subsequent, more fine-grained layers. For the non-English tasks, the same system was used after translating all documents to English.

BAMBAS (Vasconcelos et al., 2024) The proposed system for Subtask 1 does not consider the hierarchy of labels. First, text embeddings are extracted leveraging a multilingual tweets-based language model, Bernice. Next, those embeddings are used to train a separate binary classifier for each label, in a binary-relevance style, adopting independent oversampling strategies in each model.

nowhash (Chowdhury and Ptaszynski, 2024) In their submission to Subtask 1, the team starts from meme texts as input to the system and fine-tunes a Language-agnostic BERT sentence embedding (LaBSE) model on top of Flair's Transformer Document Embeddings. Further, those document vectors are then fed to a single-layer feed-forward linear classifier to obtain the prediction label.

For Subtask 2b, the proposed system operates on both meme images and texts. The architecture includes a vision transformer and XLMRoBERTa to extract effective contextual information from both modalities. Finally, the features are fused, to be passed to a single feed-forward linear layer. The architecture is fine-tuned given the task training data.

RDproj (Zhu, 2024) In their participation in Subtask 1, the team built an ensemble learning system employing a soft voting strategy. Propaganda techniques were grouped into ten subsets based on their representation in the training subset. Subsequently, one classifier including XLM-RoBerta$_{large}$ with a classification head is trained on each of these training sets. Finally, a classifier with the same architecture is used to learn a weighted average of the label's probability generated by the other classifiers.

BERTastic (Mahmoud and Nakov, 2024) For Subtask 2, the proposed architecture covers three

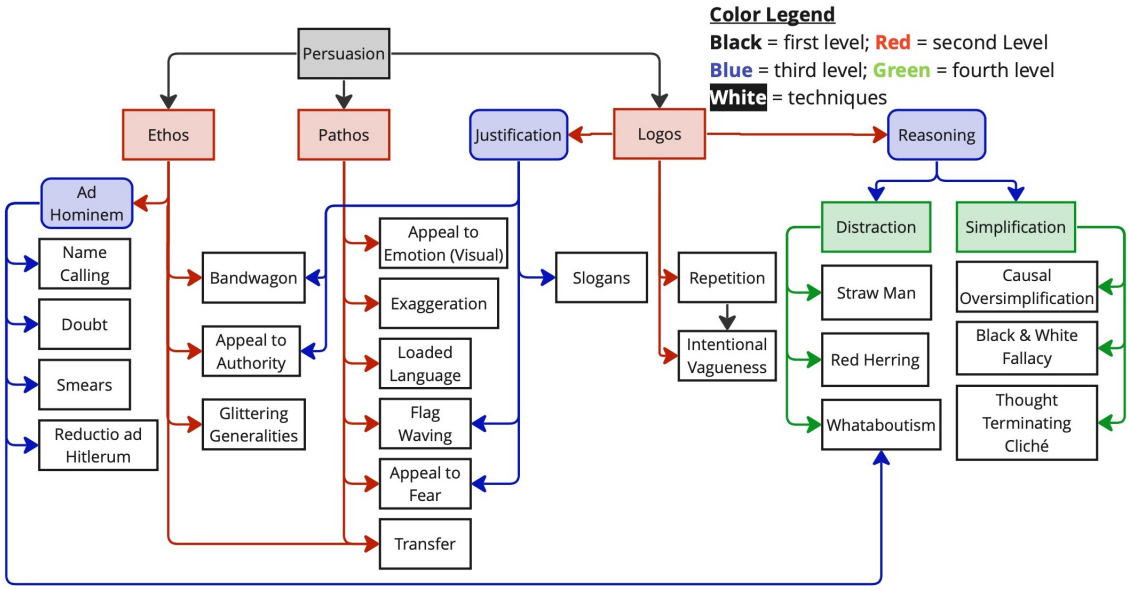


Figure 3: Persuasion techniques hierarchy.

representations of the input meme, including the image, associated text, and a generic description of the meme generated by a vision-language model. A multilingual model, MPNet, is used to extract embeddings from text elements, while a multimodal multilingual model, CLIP-ViT-B-32, is used to represent both text and image. All extracted features are fused into a single feature vector, followed by logistic regression for classification. During training, models' weights were frozen.

GreyBox (Roll and Graham, 2024) For Subtask 1, GPT 3.5 Turbo was fine-tuned in multiple stages using the training and validation datasets from all subtasks. Then, zero-shot prompting was used to generate predictions. The team also experimented with the original GPT 3.5 Turbo, Llama 2 70B Chat model, and Mistral AI's Mixtral 8x7B instruct, mixture of experts model.

SuteAlbastre (Anghelina et al., 2024) In submitting to Subtask 1, a BERT model was fine-tuned on the provided data. As for Subtask 2a: The backbone of the solutions is a BERT + ViT architecture where the BERT-based model creates embeddings from the text data while the ViT creates features from the image data. The two embeddings are concatenated and the resulting one is passed to a fully connected layer to obtain the scores for each persuasion technique. The same architecture was used for Subtask 2b, except the output of the final fully connected layer was adjusted for binary classification on whether the provided meme is propagandistic or non-propagandistic.

Pauk (Pauk and Pacheco, 2024) For Subtask 1, a student-teacher knowledge distillation approach was implemented. DeBERTa was adopted as the student model, in addition to a softened logic rule layer on top with a collection of logic rules that encode the hierarchical relationship between possible output labels. The student model then learns by both emulating the gold labels as well as the teacher's predictions that respect the hierarchy. The same knowledge distillation approach was used for Subtask 2a. However, the student model consists of DeBERTa for processing the textual content and ResNet for processing the image content with output embeddings concatenated and fed into a feed-forward network for predictions.

DUTIR938 (Yu et al., 2024) For Subtask 2b, the team developed a dual-channel model based on semi-supervised learning and model ensemble. Within the image channel, CLIP was used to extract image features from memes. Concurrently, in the text channel, diverse pre-trained language models were utilized. A concatenation and fusion process of the extracted features was applied and the resulting features were subsequently fed into a classification layer. Lastly, a two-stage soft-voting ensemble strategy was used to amalgamate the predictions of multiple models.

CLaC (Nayak and Kosseim, 2024) Similar to several other systems submitted to Subtask 1, the proposed approach was based on fine-tuning indi-

vidual language models (BERT, XLM-RoBERTa, and mBERT) and leveraging a mean-based ensemble model. Additionally, the training dataset was augmented by a relevant dataset extracted from a previous SemEval task

EURECOM (Peskine et al., 2024) The proposed system for Subtask 1 uses an ensemble of multiple models trained with different parameters. Experiments were conducted with different models (BERT, RoBERTa, DeBERTa, DistillBERT, AlBERT), different training datasets (SemEval 2024, + 2021, + PTC), different loss functions (BCE, CE, Focal, Hierarchical) and data augmentation (back translation, GPT-4-turbo augmented). The best results were obtained by leveraging the hierarchical nature of the data, by outputting ancestor classes and with a hierarchical loss. The official submission was based on the majority voting of our top-3 models for each persuasion technique.

Fralak (Laken, 2024) Different from transformer-based approaches presented so far, the system developed for Subtask 1 involved training an RNN. It was based on restructuring the labels into strings that showed the full path through the label hierarchy, and training a basic RNN that generated these strings based on the multilingual sentence embedding of the meme text. This RNN module was then incorporated into an ensemble model with 2 more models consisting of basic fully connected networks.

Snarci (Zedda et al., 2024) The system submitted to Subtask 2b involved a modular architecture that combines image and language embedding models. As image encoders, several versions of CLIP were used. Similarly, to process the textual part of memes the system resorts to several pre-trained language models (specifically TinyLlama, phi-1.5, and phi-2). The embeddings extracted from the CLIP model undergo an image embedding projection to fit a compatible size for large language models. An optional Token Merger module, inspired by the Patch Merger module proposed in vision transformers, merges tokens from image and text embeddings to focus on relevant meme aspects. This module aims to aggregate similar tokens together, regardless of their original position. To make the system more computationally efficient, freezing techniques were used to maintain the pre-trained weights of both image and language embeddings, and then Low-Rank Adaptation techniques were leveraged to fine-tune the models' weights.

SheffieldVeraAI (Grimshaw et al., 2024) For Subtask 2b, the team approached the problem by prompting and fine-tuning the large vision-language model, LLaVa. Fine-tuning was done using the multi-modal training data through LoRA training technique, however, this did not improve the model's performance. We achieved the best results prompting the baseline LLaVa model. We adapted the model to the unseen languages, by using a machine translation model, NLLB. We translated the meme transcriptions into English and used this translated text prompt with the original meme.

ShefCDTeam (Gibbons et al., 2024) The team participated in subtask 1, exploring sequence-to-sequence modeling for this task using a Flan-T5 model with sequential parameter efficient fine-tuning methods - Low-Rank Adaptation and prompt tuning.

whatdoyoumeme (Chatterjee et al., 2024) Subtask 1 was approached by fine-tuning a transformer model. The hierarchical labels for the task were integrated into the system by extending the training labels to include all ancestors. Experiments were conducted using several models like DistilBERT and mBERT but the best results were achieved with mBART. The model employed the standard classification architecture (mBART+classification head) and was trained using a BCE loss. When running the system over the non-English test sets, the documents were translated to English using the NLLB-200 model.

LomonosovMSU (Skiba et al., 2024) Two approaches were used to solve Subtask 1. 1) A generative approach involving training a generative model to generate explicit responses to questions. 2) A BERT-like approach involving training a simple fully connected network on top of a frozen pre-trained embedding model to solve the hierarchical classification task. Subtask 2 was tackled similarly to Subtask 1, but using multimodal text-to-image embedding models.

HierarchyEverywhere (Ghahroodi and Asgari, 2024) In Subtask 1, a state-of-the-art hierarchical text classification model called HPT was used. This required representing the propaganda techniques hierarchically as a directed acyclic graph. Two supplementary datasets were also added to the training. In Subtask 2a and Subtask 2b, the image

component of memes was disregarded, and only the textual content was provided to the model. Furthermore, for all the sub-tasks that are non-English, Google Translation API was used to translate them into English.

914isthebest (Li et al., 2024a) The team developed a transfer-based model for Subtask 1. For system training, the training dataset was augmented following a Chain-of-Thought-based data augmentation approach using GPT-3.5. The main classification architecture includes four RoBERTa models and one DeBERTa model initialized using different random seeds. A soft voting approach, which averages the predicted probabilities of each label from all five models, is used to predict labels. To predict non-English languages, the testing sets were translated using GPT-3.5.

LMEME (Li et al., 2024b) In Subtask 2b, the team proposed a detection system that employs a Teacher Student Fusion framework. Initially, a large language model serves as the teacher, engaging in abductive reasoning on multimodal inputs to generate background knowledge on persuasion techniques, assisting in the training of a smaller downstream model. The student model adopts CLIP as an encoder for text and image features, incorporating an attention mechanism for modality alignment.

McRock (Siino, 2024) The team approached Subtask 1 by prompting an instruction-tuned large language model called Mistral-7B-Instruct-v0.2. The prompt used included both the definitions of all 20 techniques targeted by the subtask, a short instruction on the task to perform, and the sample to predict on. The post-processed model's outputs were then submitted to the task's leaderboard.

BCAmirs (Abaskohi et al., 2024) The team participated in all subtasks but mainly focused on Subtask 2a. GPT-4 was used to generate a descriptive caption of the meme. The caption is then combined with the meme text before being passed to a RoBERTa model. A vision encoder utilizing a pre-trained vision transformer model (CLIP-ViT), is used to encode and analyze meme images. Finally, a multi-layer perceptron classifier takes the combined visual and textual representations and classifies the meme. The RoBERTa and MLP classifiers are fine-tuned, while CLIP remains frozen.

They conducted a series of experiments exploring different methods of combining the textual and visual data: *text-only* (Vicuna-1.5, BERT, RoBERTa), *image-only* (LLaVa without textual input), *text + image* (VisualBERT, ConcatRoBERTa, LLaVa-1.5), *text + caption + image* (LLaVa-1.5, Vicuna-1.5, VisualBERT, ConcatRoBERTa). Experiments were conducted using LLaVa and GPT-4 generated captions with GPT-4 captions showing consistently better results.

Puer (Dao et al., 2024) The team participated in Subtask 1 on the English test data with a detection system based on RoBERTa, using Roberta-large, which was fine-tuned on a corpus of social media posts. They conducted extensive parameter tuning over the dev set to identify an optimal threshold, epoch, etc. Finally, They compare the performances of other different deep learning model architectures, such as BERT, ALBERT, and XLM-RoBERTa, on multilingual detection of persuasion techniques in memes.

Hidetsune (Takahashi, 2024) The team approached Subtask 2b with a text-only classical NLP solution using SpacyV3 textcat_multilabel classification architecture. The model was trained on the official dataset for Subtask 2b, combined with additional data from Kaggle consisting of non-propagandistic tweets. The team participated in all languages included in Subtask2b by translating non-English text into English and applying the same model for text classification.

UMUTeam (pan et al., 2024) The team participated in all subtasks of the competition focusing only on English data. In Subtask 1 the team fine-tuned the RoBERTa-large model using an epoch-based evaluation strategy. In Subtasks 2a and 2b, they again used RoBERTa-large as their classification model but trained it by combining the textual content of a meme with image descriptions extracted using LlaVa.

MagnumJUCSE (Khurshid and Das, 2024) The team participated in Subtask 1 in all languages. They participated in the subtask with a node-level hierarchical classification system consisting of four phases: data denoising, feature generation, node-level classifier training, and finally inference. They first clean the data, then generate features using pre-trained sentence transformers, afterwards they predict whether an example belongs to a given node or not using SVM (Support Vector Machine). Finally, inference is done in a top-down fashion by selecting the most suitable depth for the prediction

results, based on the decision probabilities of the classifier at each node.

IUSTNLPLAB (Osoolian et al., 2024) The team addresses Subtask 1 on the English dataset. Their study focused on fine-tuning language models using the training dataset, including BERT, GPT-2, and RoBERTa, with GPT-2 showing the best performance for the task. Additionally, they used data on persuasion techniques from Semeval 2023 Task 3 increasing the training data with 3,445 new samples, however, this approach did not yield discernable improvements. Finally, the participants adjusted the prediction threshold which lead to a noticeable improvement in model performance.

IITK (Chikoti et al., 2024) The team participated in all three Subtasks in every language. Subtask 1: they presented an approach to meme classification based on HypEmo (pre-trained hyperbolic embeddings) and emotion prediction through a multi-task learning framework, incorporating auxiliary tasks, including masked language modeling (MLM) and class definition prediction to enhance the understanding of emotional concepts. The predictions from HypEmo and the Fine-grained class-definition-based model are merged for the final prediction. Subtask 2a: the team experiments with an ensemble of HypEmo and the class definition-based multi-task learning model for the textual content of the meme and using the CLIP model embeddings from the visual content of the meme. Subtask 2b: the team uses a fusion approach, concatenation pre-trained BERT-base model for textual features and CNN model for visual features. They use weighted binary cross entropy as a loss function due to the dataset imbalance.

NLPNCHU (Guo and Fan, 2024) The team participated in all three Subtasks in every language. They explored various finetuning techniques and classification strategies, such as data augmentation, problem transformation, and hierarchical multi-label classification strategies. In Subtasks 1 and 2a, they explored different classification strategies: Global Classifier (GC), Stacking + GC, and Stacking + Local Classifier per Level (LCL), combined with Distribution-Balanced Loss (DBL) loss to address the long-tail distribution of the data. For Subtask 1 the team compared the performance of XLM-RoBERTa and XLM-RoBERTa-Twitter to asses the impact of domain-specific pre-training. For Subtask 2a the team used XLM-RoBERTa and XLM-RoBERTa-Twitter for extracting textual features and CLIP for extracting visual features combining them through Feature-wise Linear Modulation (FIM), these two encoders encode to obtain a representation embedding vector containing both image and text For Subtask 2b the team employed the same strategy as Subtask 2a applied to a binary classification setting.

SemEval-2024 Task 5: Argument Reasoning in Civil Procedure

Lena Held[1] and **Ivan Habernal**[2]
Trustworthy Human Language Technologies
[1] Department of Computer Science, Technical University of Darmstadt
[2] Department of Computer Science, Paderborn University
lena.held@tu-darmstadt.de, ivan.habernal@uni-paderborn.de
www.trusthlt.org

Abstract

This paper describes the results of SemEval-2024 Task 5: Argument Reasoning in Civil Procedure, consisting of a single task on judging and reasoning about the answers to questions in U.S. civil procedure. The dataset for this task contains question, answer and explanation pairs taken from *The Glannon Guide To Civil Procedure* (Glannon, 2018). The task was to classify in a binary manner if the answer is a correct choice for the question or not. Twenty participants submitted their solutions, with the best results achieving a remarkable 82.31% F_1-score. We summarize and analyze the results from all participating systems and provide an overview over the systems of 14 participants.

1 Introduction

"Arguing a legal case is an essential skill that aspiring lawyers must master. This skill requires not only knowledge of the relevant area of law, but also advanced reasoning abilities, such as using analogy arguments or finding implicit contradictions." – (Bongard et al., 2022)

In order to test these abilities, we organized the SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. By basing our dataset on an established textbook in the domain of U.S. civil procedure (*The Glannon Guide To Civil Procedure*, (Glannon, 2018)), we ensure that we can leverage the high quality and refined content aimed at law students to create a challenging task in the competition. The book follows the philosophy, that learning about civil procedure can be achieved by reading about a given topic and answering questions afterwards. Therefore, each chapter is accompanied by a set of hard reasoning problems formulated as multiple-choice questions. As a teaching resource, the book provides a thorough analysis for each answer candidate. This enables the student to learn by example.

We frame our task in a simple manner: classifying whether the given answer is a correct solution to the question or not. With this task, we want to put the legal reasoning capabilities of various state-of-the-art models to the test and provide a reliable benchmark.

2 Related work

As the task is based upon our previous paper (Bongard et al., 2022), we refer to the detailed related work section in there. In a nutshell, legal question answering is an inherently difficult task because it requires both reasoning skills and expertise. Legal question datasets in NLP are scarce and vary in terms of the specific topics covered, such as the U.S. Multistate Bar Examination (Fawei et al., 2016), Tax Law (Holzenberger et al., 2020), and Japanese Bar Exams (Kano et al., 2019; Rabelo et al., 2022). Although existing datasets focus on finding the correct answer to the question posed, the reasoning behind a correct or incorrect answer is often ignored. More recently, LLMs have found their way into legal question answering, demonstrating their potential in this area (Katz et al., 2023) by solving complex legal questions at a level comparable to humans. But these circumstances also highlight the need for appropriate tasks to evaluate such systems (Guha et al., 2023).

3 Dataset

The dataset was collected by parsing *The Glannon Guide To Civil Procedure* (Glannon, 2018) which was done in our previous work (Bongard et al., 2022). The details of the data collection and baseline methods are also outlined there. Instead of treating the questions from the book as multiple choice queries, we decided to pair each answer with its question and attach a binary label for a correct or incorrect conclusion. Because there are usually multiple incorrect answers to a question, the dataset is highly imbalanced towards incorrect answers. A question can either be a stand-alone sentence or in cloze text form. To make the context

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 2027–2038

Question 7. A switch in time. Yasuda, from Oregon, sues Boyle, from Idaho, on a state law unfair competition claim, seeking $250,000 in damages. He sues in state court in Oregon. Ten days later (before an answer is due in state court), Boyle files a notice of removal in federal court. Five days after removing, Boyle answers the complaint, including in her answer an objection to personal jurisdiction. Boyle's objection to personal jurisdiction is

Answer not waived by removal. The court should dismiss if there is no personal jurisdiction over Boyle in Oregon, even though the case was properly removed.

Solution 1

Analysis D is the correct answer. Boyle has not waived his objection to personal jurisdiction. If the federal court lacks jurisdiction over Boyle, it should dismiss the case, even though it was properly removed.

Complete Analysis There are so many ways to go astray on this issue [...].

Introduction My students always get confused about the relationship between removal to federal court and personal jurisdiction [...].

Figure 1: Example data point

of most of the questions clear, there is an introduction text which provides background information to the question. In addition, Glannon has written further explanatory texts which justify why the answer was a correct choice or not. Each data point in the dataset consists of *question, answer candidate, solution, analysis (answer), complete analysis (all answers to the question), introduction*. An example data point is presented in Figure 1.

However, the dataset version used in the competition differs slightly from the original version. To correct errors in the initial version of the dataset, we removed a mistakenly included chapter of the book. Additionally, we corrected two instances in which the explanation text was missing. Although the dataset size changed, the partitions are still based on the paradigm used in (Bongard et al., 2022), resulting in a training partition (666), development partition (84), and testing partition (98). The *rational data split* is meant to sort questions which appear later in each chapter into the test set, assuming that these questions are harder to answer than earlier ones. To conceal the labels in the test set, we eliminated both fields *label* and *analysis* in that partition.

3.1 Potential question leakage from dev to test

When splitting the dataset partitions, we created some unwanted potential leakage. In particular, some questions that appear in the test partition may have already been part of the development partition with a different answer candidate. This occurred because each partition should contain questions from each chapter and data points were not considered as questions with multiple answer candidates, but rather as question-answer pairs. Because some dataset requests had already been answered, we chose not to readjust the partitioning. The training partition is not affected by this. About 27 of 98 data points in the test partition are affected and due to the small size of the dataset, we chose not the remove the data points either.

Instead, we take this opportunity to analyze if the behavior of the participating systems differs in regards to the leaked questions. The details of this additional analysis are presented in section 6.2. However, a future version of the dataset will contain a modified split that fixes the issue.

4 Task description

Reasoning is still one of the hardest task state-of-the-art models and techniques can face. Simply understanding language is certainly not enough to understand expert legal questions, much less answer them correctly. The task is meant to probe the capacity of methods for understanding complex legal topics and applying them in exemplary scenarios. However, to avoid over-complicating the output and evaluation, the task is formulated as a simple yes or no question. By default this approach also makes the task harder, because there is no option to find one correct answer by process of elimination. The task remains the same as introduced by Bongard et al. (2022):

Task Given a question with a possible correct answer and a short introduction to the topic of

the question, identify if the answer candidate is correct or incorrect.

Although systems may use the analysis that is provided in the training and development partitions for enhancement, they should be able to produce a prediction based on introduction, question and answer candidate alone.

4.1 Evaluation methods

Due to the simplicity of the task itself, we consider standard metrics to be best suited to evaluate the submissions. We calculate the macro F_1-score to account for the dataset imbalance between correct and incorrect answers. We evaluate the accuracy as well as an additional point of comparison. The F_1-score is the relevant evaluation metric for the competition ranking.

As a baseline, we provide a simple majority baseline which predicts each answer as incorrect and achieves an F_1-score of 42.69%.

4.2 Organization

We setup the competition on the CodaLab platform.[1] Participants needed to register first and acquire the dataset by filling out the required form as agreed with the publisher of the book[2]. We sent out the training and development partitions of the dataset first. The practice phase of the competition was officially accessible from November 28th, 2023 to allow participants to get accustomed to the submission platform and upload their scores for the development set. The test partition was sent out on January 9th, 2024 via email to those who had previously requested the dataset. Between January 10th, 2024 and February 1st, 2024 (00:00:00 UTC), participants could upload up to 5 submissions in total. After the end of the evaluation phase, participants could still upload contrastive runs in the post-evaluation phase with the same evaluation script.

5 Participant systems

During the competition period, we received 59 requests for the dataset. Of the 55 participants who registered on the CodaLab platform, 20 submitted results in the evaluation phase. We summarize and evaluate the 14 teams that submitted system papers.

[1] https://codalab.lisn.upsaclay.fr/competitions/14817

[2] https://github.com/trusthlt/legal-argument-reasoning-task

Rank	Participant	Acc.	F_1
1	HW-TSC	0.8673	0.8231
2	MAINDZ	0.8265	0.7747
3	SU-FMI	0.8367	0.7728
4	qiaoxiaosong	0.8163	0.7644
5	UTSA-NLP	0.7959	0.7315
6	kubapok	0.7857	0.6971
7	LegalSense	0.7449	0.6599
8	hrandria	0.6939	0.6327
9	Yuan_Lu	0.6327	0.6000
10	PengShi	0.6735	0.5910
11	Mistral	0.5714	0.5597
12	Hwan_Chang	0.5918	0.5556
13	kriti7	0.6020	0.5511
14	woody	0.6633	0.5510
15	odysseas_aueb	0.6122	0.5143
16	SCaLAR Group, NITK Surathkal	0.6224	0.4966
17	lhoorie	0.5000	0.4957
18	yms	0.7245	0.4827
19	U_201060	0.6633	0.4503
20	langml	0.4490	0.4375
21	majority baseline	0.7449	0.4269

Table 1: Official Leaderboard, counting the last submission made by a participant.

In addition to the descriptions, we present a brief summary of the key features of the proposed systems in Table 3.

5.1 Leaderboard results

We allowed participants to make up to 5 submissions in the evaluation phase to encourage them to try out several approaches. For the official leaderboard, which is taken from CodaLab, only the last valid submission is counted, resulting in the ranking shown in Table 1. We have also created a leaderboard that counts the best submission instead of the last one. This leaderboard variant is shown in Table 2. The differences between the leaderboard rankings are minimal. Both leaderboards are available on the competition webpage[3].

5.2 System descriptions

The systems mostly rely on established LLMs like GPT-4 (OpenAI, 2023), Llama (Touvron et al., 2023a) or Llama 2 (Touvron et al., 2023b), Mistral (Jiang et al., 2023) or Mixtral (Jiang

[3] https://trusthlt.github.io/semeval24/

Rank	Participant	Acc.	F_1
1	HW-TSC	0.8673	0.8231
2	MAINDZ	0.8265	0.7747
3	SU-FMI	0.8367	0.7728
4	qiaoxiaosong	0.8163	0.7644
5	UTSA-NLP	0.8061	0.7341
6	kubapok	0.7857	0.6971
7	LegalSense	0.7449	0.6599
8	hrandria	0.6939	0.6327
9	PengShi	0.6837	0.6166
10	Yuan_Lu	0.6327	0.6000
10	Hwan_Chang	0.6735	0.6000
12	Mistral	0.5714	0.5597
13	kriti7	0.6020	0.5511
14	woody	0.6633	0.5510
15	SCaLAR Group, NITK Surathkal	0.6429	0.5238
16	odysseas_aueb	0.6122	0.5143
17	lhoorie	0.5000	0.4957
18	yms	0.7245	0.4827
19	U_201060	0.6633	0.4503
20	langml	0.4490	0.4375
21	majority baseline	0.7449	0.4269

Table 2: Leaderboard, counting the best submission made by a participant.

et al., 2024), Zephyr (Tunstall et al., 2023) or Flan-T5 (Longpre et al., 2023). Other popular models are Legal-BERT (Chalkidis et al., 2020), RoBERTa (Liu et al., 2019), Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020). Many teams explore different strategies to prompt the LLMs, for instance using Chain-of-Thought (Wei et al., 2022).

Rank 1: HW-TSC – Self-Eval? A Confident LLM System for Auto Prediction and Evaluation for the Legal Argument Reasoning Task (Zhao et al., 2024) This team uses different GPT-4 prompt designs and strategies alongside a self evaluation approach leveraging a confidence score. Their best-performing system remodels the task into a multiple-choice question answering task and uses an ensemble of 3 runs. The authors' experiments show that prompting the LLM for a confidence score improves the performance in all tested settings. Their results also highlight that remodeling the task into a multiple-choice question answer task improves the performance significantly.

Rank 2: MAINDZ – CLUEDO - Choosing Legal oUtcome by Explaining Decision through Oversight (Benedetto et al., 2024) This team took an interesting approach by employing a two-stage decision process. In the first step, an ensemble of three models is fine-tuned with all available information (introduction, questions, answer cast as multiple-choice task) and not only generates the correct predictions, but also the explanations. In the second step, these generated candidates are evaluated by another zero-shot system (a 'detective') which chooses the final solution (given the labels and the explanations).

Rank 3: SU-FMI – From BERT Fine-Tuning to LLM Prompt Engineering - Approaches in Legal Argument Reasoning (Krumov et al., 2024) The authors experimented with a large number of approaches, starting with fine-tuning BERT-based models, adding external fine-tuning data, over to utilizing commercial LLMs with prompt engineering. The best results were achieved by utilizing GPT-4 and legal prompt engineering (prompts tailored for legal reasoning tasks). This team also provides a thorough comparison with other, partly open-source models.

Rank 5: UTSA-NLP – Prompt Ensembling for Argument Reasoning in Civil Procedures with GPT4 (Schumacher and Rios, 2024) This team uses the analysis part as a Chain-of-Thought mechanism in in-context learning. In particular, they prompt GPT-4 which, given the intro, question, and the answer candidate at test time, also generates the analysis part and the final label. The final system is an ensemble model combining several variants of the base models. The authors also provide an error analysis, showing that longer introductions tend to confuse the models.

Rank 7: NLP at UC Santa Cruz – Legal Answer Validation using Few-Shot Multi-Choice QA (Pahilajani et al., 2024) This team analyzed several fine-tuning strategies based on BERT models, or the effects of integrating additional CaseHold data, but concludes that multi-choice QA few-shot prompting on GPT-4 was the most effective method in their experiments.

Rank 9: 0x.Yuan – Enhancing Legal Argument Reasoning with Structured Prompts (Lu and Kao, 2024) The team investigates several prompting strategies on Mixtral-8x7B in a zero-shot man-

ner which make use of established legal reasoning methodologies like the IRAC (Issue, Rule, Application, Conclusion) analysis. The authors note that prompt designs tailored to legal reasoning methods outperform Chain-of-Thought strategies and direct prompting.

Rank 10: YNU-HPCC – Regularized Legal-BERT for Legal Argument Reasoning Task in Civil Procedure (Shi et al., 2024) The approach by this team employs fine-tuning of Legal-BERT and other BERT models and overcomes the input limitations by applying sliding window approaches. On top of comparing several losses (Cross-Entropy, Focal, Dice), they also compare the use of Regularized Dropout and Supervised Contrastive Learning for data augmentation and imbalances.

Rank 11: Mistral – Mistral 7B for argument reasoning in Civil Procedure (Siino, 2024) This team tested the pre-trained LLM Mistral-7B in a zero-shot prompting manner to classify a given question-answer pair.

Rank 13: Transformers – Legal Argument Reasoning Task in Civil Procedure using RoBERTa (Singhal and Bedi, 2024) The approach proposed by this team fine-tunes a pre-trained RoBERTa model with all input fields available in the training data and further uses minority sampling to counter the dataset imbalances.

Rank 14: ignore – A Legal Classification Model with Summary Generation and Contrastive Learning (Sun and Zhou, 2024) The team uses a Legal-BERT classifier with a contrastive learning approach. They additionally shorten the introduction text by summarizing it with GPT-3.5 and augment the training data by concatenating parts of the input in different ways. The authors note that generative summarization proves feasible to handle the introduction text and the contrastive loss improves the robustness of the model.

Rank 15: Archimedes-AUEB – LLM explains Civil Procedure (Chlapanis et al., 2024) This team proposes extending the training data by synthetic data generated by GPT-3, where the generated data resemble Chain-of-Thought reasoning. The authors also fine-tune a student model, an open-source Llama-2-7b, with QLoRA and provide an expert-based analysis, which reveals some shortcomings in explanations of the models.

Rank 16: SCaLAR NITK – Towards Unsupervised Question Answering system with Multi-level Summarization for Legal Text (Prabhu et al., 2024) The team tried various approaches using Word2Vec, GloVe and Legal-BERT embeddings to identify the most likely answer in a multiple-choice setup based on similarity scores. Additionally, they employ a segment-wise summarization of the introduction text with T5 and investigate the differences in similarity scores between the summarized and original input. The approach relies on open-source models and is reproducible.

Rank 17: eagerlearners – The Legal Argument Reasoning Task in Civil Procedure (Sabzevari et al., 2024) This team experimented with different designs for prompting GPT-3.5, Gemini and Copilot in a zero-shot manner. In additional experiments, the authors find that among some BERT-family models, a fine-tuned Legal-BERT exhibits the best potential, outperforming Longformer and Big Bird.

Rank 18: DUTh – A multi-task learning approach for the Legal Argument Reasoning Task in Civil Procedure (Maslaris and Arampatzis, 2024) This team compared the Legal-BERT model with a multi-task Flan-T5 model, which eventually performed on par. The authors relied mostly on fully open-source models and make their approach reproducible.

6 Analysis

6.1 Error analysis

We take a closer look how individual instances in the test set were classified. For this, we cluster the instances by the chapter they appear in and sort the chapters by the average performance (see Figure 2). With the goal of identifying the questions that were more challenging for the systems to answer, we cross-check the chapter titles and content of the best and worst-performing chapters. Chapters 6, 12, and 7 were the best-performing and cover the topics "More Personal Jurisdiction: General In Personam Jurisdiction and In Rem Jurisdiction", "Two Ways to Run a Railroad: Substance and Procedure After York, Byrd, and Hanna" and "More than an Afterthought: Long-arm Statutes as a Limit on Personal Jurisdiction". Legal expertise would be required to carefully assess why some chapters appear more difficult than others. Throughout our analysis, we could not identify a clear common

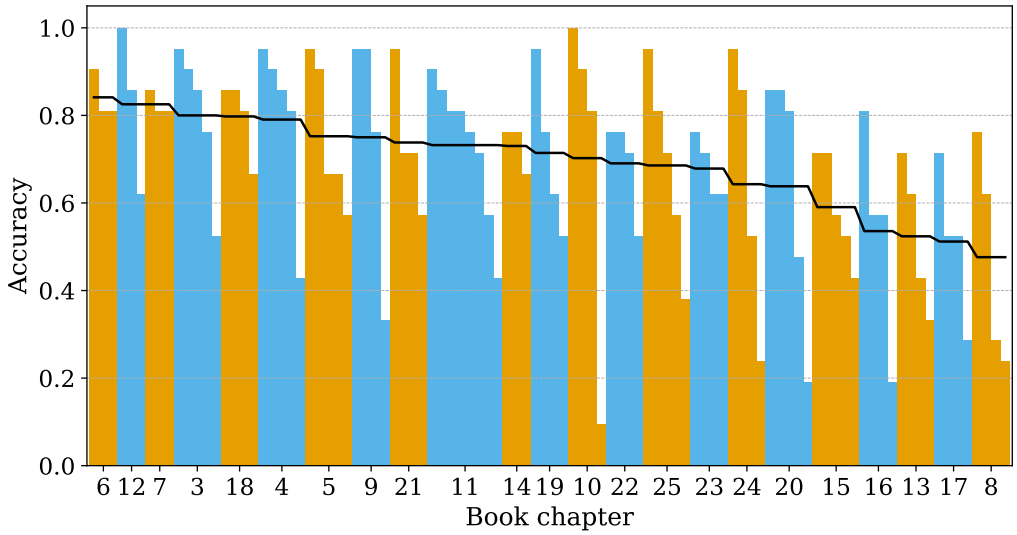


Figure 2: Prediction accuracy of all systems on all questions individually, grouped by the chapter the questions appear in *The Glannon Guide To Civil Procedure*. The line indicates the average accuracy per chapter. The alternating colors serve to delimit the individual chapters.

factor for difficult and easy instances. This can be attributed to the small sample size of the test partition and the carefully designed questions. Please refer to Table 5 for a full list of chapter titles.

Another important distinction is between question-answer pairs with a correct answer and those with an incorrect answer. As expected, because of the imbalance of the dataset, correct answers were much harder to classify correctly, as shown in Figure 3 (highlighted in green). On average, only 48.76% of these instances were classified correctly by all participants. For incorrect answers, 76.25% were classified correctly.

6.2 Potentially leaked data points

Furthermore, we want to investigate the impact of our potentially leaked data points. We compare the performance on non-leaked questions to that on potentially leaked questions in Figure 3 (indicated by a red border) and find that the performance remains almost identical for incorrect answers (76.69% for leaked vs. 76.10% for non-leaked), but shows a slight increase for correct answers (53.57% for leaked vs. 46.50% for non-leaked).

Table 4 also displays the difference in the final score that would result from removing potentially leaked data points for each participant. While the ranking may change for some teams, the gains and losses are minimal and do not follow a discernible pattern.

All in all, we could not detect a strong impact of

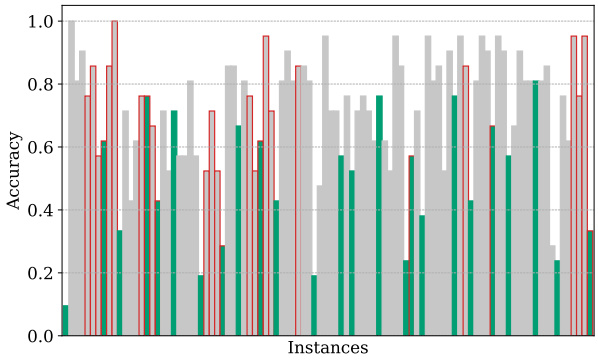


Figure 3: Prediction accuracy of all instances in the test set. Green instances mark questions-answer pairs with a true answer. Indicated by red boxes are instances that could have potential leakage of the question from the dev set.

the potentially leaked data points. This could also be due to the very limited use of fine-tunining or training with the provided data, since many models simply use zero-shot prompting or similar methods that do not require the training data at all.

6.3 Findings

The best-performing systems all use GPT-4, either with a double-checking mechanism (prompting more than once), tailoring the prompt to a legal reasoning method, or using ensembling to achieve optimal results. Domain-specific models, such as the popular Legal-BERT, which were explored in several approaches, are consistently outperformed by systems using GPT-4 and could not demonstrate

their advantages. The authors of some systems also noted that task performance improved when the task was remodeled as a multiple-choice task. Although this was not prohibited, it undermines the idea of the task and should be taken into account in a potential future iteration. Lastly, additional data was rarely used and did not contribute to the best results. Although the focus of the best submissions was on leveraging the power of LLMs, the techniques used to acquire a label from the prompts were creative, diverse and tailored to the legal domain.

7 Conclusion

In this paper we presented an overview of Task 5 of the SemEval-2024 competition, a task on argument reasoning in civil procedure. The dataset and the problems related to data leakage due to partitioning were briefly outlined. The submitted systems were described and summarized, and insights into the achieved results were provided. The submitted solutions indicate that LLMs, specifically GPT-4, are surprisingly decent in handling argument reasoning in civil procedure. Although Legal-BERT and other older domain-specific models can still solve the task to some extent, they are outperformed by a significant margin. The average performance of older or simpler techniques also suggests that this task is a suitable benchmark for evaluating legal reasoning in civil procedure. Although the top-performing systems still have room for improvement, the submitted solutions demonstrate that performance can be enhanced using various techniques. This task is far from solved. A future iteration of this competition could also utilize the mostly unused *analysis* field. This could alleviate the dataset's shortcoming of lacking traceable reasoning steps in the solution to further boost the emphasis on the reasoning aspect of the task.

Limitations

In theory, the dataset should not have leaked to a large language model yet, because the book is not freely available online. Consequently, the dataset should contain mostly new and unseen questions for the NLP community, while also having limited risk of leakage into a large language model. However, especially because of the use of closed LLMs and the lack of knowledge about the training corpora used for them, we can not be entirely sure that our dataset has not been seen by the LLMs used in the systems.

Although some of the answers to the questions can be argued about and might even be outdated in terms of applicable laws and statutes (the basis for the dataset is the 4th edition of the book), we can consider them correct, because they were answered by an expert – the author of the book.

Acknowledgements

We would like to thank John Glannon and Aspen Publishing for their support. This work has been funded by the German Research Foundation as part of the ECALP project (HA 8018/2-1).

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Irene Benedetto, Alkis Koudounas, Lorenzo Vaiani, Eliana Pastor, Luca Cagliero, and Francesco Tarasconi. 2024. MAINDZ at SemEval-2024 task 5: CLUEDO - Choosing Legal oUtcome by Explaining Decision through Oversight. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 986–994, Mexico City, Mexico. Association for Computational Linguistics.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Odysseas Chlapanis, Ion Androutsopoulos, and Dimitrios Galanis. 2024. Archimedes-AUEB at SemEval-2024 task 5: LLM explains civil procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1617–1632, Mexico City, Mexico. Association for Computational Linguistics.

Biralatei Fawei, Adam Wyner, and Jeff Pan. 2016. Passing a USA national bar exam: a first corpus for experimentation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3373–3378, Portorož, Slovenia. European Language Resources Association (ELRA).

Joseph W Glannon. 2018. *The Glannon Guide To Civil Procedure: Learning Civil Procedure Through*

Multiple-Choice Questions and Analysis, 4th edition. Aspen Publishing.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. In *Proceedings of the Natural Legal Language Processing Workshop 2020 co-located with the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2020)*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts.

Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2019. COLIEE-2018: Evaluation of the competition on legal information extraction and entailment. In *New Frontiers in Artificial Intelligence*, volume 11717 of *Lecture Notes in Computer Science*, pages 177–192, Cham. Springer International Publishing.

Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. 2023. GPT-4 passes the Bar Exam.

Kristiyan Krumov, Svetla Boytcheva, and Ivan Koytchev. 2024. SU-FMI at SemEval-2024 task 5: From BERT fine-tuning to llm prompt engineering - approaches in legal argument reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1662–1668, Mexico City, Mexico. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan collection: Designing data and methods for effective instruction tuning.

Yu-An Lu and Hung-Yu Kao. 2024. 0x.Yuan at SemEval-2024 task 5: Enhancing legal argument reasoning with structured prompts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 385–390, Mexico City, Mexico. Association for Computational Linguistics.

Ioannis Maslaris and Avi Arampatzis. 2024. DUTh at SemEval 2024 task 5: A multi-task learning approach for the legal argument reasoning task in civil procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1042–1046, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report.

Anish Pahilajani, Samyak Rajesh Jain, and Devasha Trivedi. 2024. NLP at UC Santa Cruz at SemEval-2024 task 5: Legal answer validation using few-shot multi-choice QA. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1298–1303, Mexico City, Mexico. Association for Computational Linguistics.

M Manvith Prabhu, Haricharana Srinivasa, and Anand Kumar M. 2024. SCaLAR NITK at SemEval-2024 task 5: Towards unsupervised question answering system with multi-level summarization for legal text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 193–199, Mexico City, Mexico. Association for Computational Linguistics.

Juliano Rabelo, Randy Goebel, Mi-Young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2022. Overview and discussion of the competition on legal information extraction/entailment (COLIEE) 2021. *The Review of Socionetwork Strategies*, 16(1):111–133.

Hoorieh Sabzevari, Mohammadmostafa Rostamkhani, and Sauleh Eetemadi. 2024. eagerlearners at SemEval2024 task 5: The legal argument reasoning task in civil procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 909–913, Mexico City, Mexico. Association for Computational Linguistics.

Dan Schumacher and Anthony Rios. 2024. Team UTSA-NLP at SemEval 2024 task 5: Prompt ensembling for argument reasoning in civil procedures with GPT4. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1283–1290, Mexico City, Mexico. Association for Computational Linguistics.

Peng Shi, Jin Wang, and Xuejie Zhang. 2024. YNU-HPCC at SemEval-2024 task 5: Regularized Legal-BERT for legal argument reasoning task in civil procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 743–748, Mexico City, Mexico. Association for Computational Linguistics.

Marco Siino. 2024. Mistral at SemEval-2024 task 5: Mistral 7B for argument reasoning in civil procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 155–162, Mexico City, Mexico. Association for Computational Linguistics.

Kriti Singhal and Jatin Bedi. 2024. Transformers at SemEval-2024 task 5: Legal argument reasoning task in civil procedure using RoBERTa. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 956–959, Mexico City, Mexico. Association for Computational Linguistics.

Binjie Sun and Xiaobing Zhou. 2024. ignore at SemEval-2024 task 5: A legal classification model with summary generation and contrastive learning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 517–522, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Xiaofeng Zhao, Xiaosong Qiao, Kaiwen Ou, Min Zhang, Su Chang, Mengyao Piao, Yuang Li, Yinglu Li, Ming Zhu, Yilun Liu, Feiyu Yao, shimin tao, Hao Yang, and Yanfei Jiang. 2024. HW-TSC at SemEval-2024 task 5: Self-eval? a confident llm system for auto prediction and evaluation for the legal argument reasoning task. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1817–1821, Mexico City, Mexico. Association for Computational Linguistics.

A Participants systems

#	Team	LLM	Prompting	Fine-tuning	Inputs	+Data	MC
1	HW-TSC	GPT-4	custom	–	Q, A, E	–	✓
2	MAINDZ	Flan T5 XXL, Llama 13B, Zephyr 7B, Mistral 7B, GPT-4	zero-shot	✓	Q, A, E	–/✓	✓
3	SU-FMI	GPT-4	custom	–	Q, A, E	–	–
5	UTSA-NLP	GPT-4	CoT	–	Q, A, E	–	–
7	UC Santa Cruz	GPT-4	zero-shot	–	Q, A, E	–/✓	✓
9	0x.Yuan	Mixtral-8x7B	CoT	–	Q, A, E	–	–
10	YNU-HPCC	Legal-BERT	–	✓	Q, A, E	–	–
11	Mistral	Mistral 7B Instruct	zero-shot	–	Q, A	–	–
13	Transformers	RoBERTa	–	✓	Q, A, E, An.	–	–
14	ignore	Legal-BERT, GPT-3.5	–	✓	Q, A, E, An.	–	–
15	Archimedes-AUEB	GPT family, Llama2 7B	CoT	✓	Q, A, E	–	–
16	SCaLAR NITK	Legal-BERT, T5	–	–	Q, A, E	–	✓
17	eagerlearners	Longformer, Big Bird, Legal-RoBERTa, GPT-3.5, Gemini, Copilot	CoT, zero-shot	✓	Q, A, E	–	–
18	DUTh	Legal-BERT, Flan T5	–	✓	Q, A, E	–	–

Table 3: Summarized features of the submitted systems.

B Leaderboard accounting for leaked data points

Rank	Participant	F_1	Diff
1	SU-FMI	0.8143	0.0415
2	HW-TSC	0.7829	-0.0403
2	MAINDZ	0.7829	0.0082
4	qiaoxiaosong	0.7535	-0.0109
5	UTSA-NLP	0.7464	0.0149
5	kubapok	0.7464	0.0493
7	hrandria	0.6048	-0.0279
8	LegalSense	0.6019	-0.0580
8	odysseas_aueb	0.6019	0.0875
10	Mistral	0.5824	0.0227
11	Hwan_Chang	0.5750	0.0195
12	PengShi	0.5594	-0.0316
13	kriti7	0.5177	-0.0335
14	Yuan_Lu	0.5127	-0.0873
15	yms	0.5071	0.0244
16	lhoorie	0.5007	0.0050
17	woody	0.4970	-0.0541
18	SCaLAR Group, NITK Surathkal	0.4779	-0.0187
19	langml	0.4510	0.0135
20	majority baseline	0.4320	0.0051
21	U_201060	0.4283	-0.0219

Table 4: Performance of the systems on data points that have not potentially leaked from dev, compared to the original score with potentially leaked data points.

C The Glannon Guide to Civil Procedure – Chapters

Chapter	Title
3	Federal Claims and Federal Cases
4	Removal Jurisdiction: The Defendant Chooses the Forum
5	Personal Jurisdiction: Myth and Minimum Contact
6	More Personal Jurisdiction: General In Personam Jurisdiction and In Rem Jurisdiction
7	More than an Afterthought: Long-arm Statutes as a Limit on Personal Jurisdiction
8	Home and Away: Litigating Objections to the Court's Jurisdiction
9	Due Process and Common Sense: Notice and Service of Process
10	Venue and Transfer: More Limits on the Place of Suit
11	State Law in Federal Courts: Basics of the Erie Doctrine
12	Two Ways to Run a Railroad: Substance and Procedure After York, Byrd, and Hanna
13	The Scope of the Action: Joinder of Claims and Parties Under the Federal Rules
14	Of Hooks and Nuclei: Supplemental Jurisdiction over State Law Claims
15	Sufficient Allegations: Pleading Under the Federal Rules
16	Change over Time: Amending the Pleadings Under Rule 15
17	Never Forget Rule 11: Representations to the Court
18	Technicalities, Technicalities: Pre-answer Motions Under the Federal Rules
19	Probing to the Limits: The Scope of Discovery Under the Federal Rules
20	The Basic Tools of Discovery in Federal Court
21	Dispositive Motions: Dismissal for Failure to State a Claim and Summary Judgment
22	Judgment as a Matter of Law in the Federal Courts
23	Second Time Around: The Grounds and Procedure for Motions for New Trial
24	The Quest for Finality: Claim Preclusion Under the Second Restatement of Judgments
25	Collateral Estoppel, Issue Preclusion, Whatever

Table 5: Chapter titles

SemEval-2024 Task 3: Multimodal Emotion Cause Analysis in Conversations

Fanfan Wang[1], **Heqing Ma**[1], **Jianfei Yu**[1*], **Rui Xia**[1*], **Erik Cambria**[2]
[1] School of Computer Science and Engineering,
Nanjing University of Science and Technology, China
[2] Nanyang Technological University, Singapore
{ffwang, hqma, jfyu, rxia}@njust.edu.cn, cambria@ntu.edu.sg

Abstract

The ability to understand emotions is an essential component of human-like artificial intelligence, as emotions greatly influence human cognition, decision making, and social interactions. In addition to emotion recognition in conversations, the task of identifying the potential causes behind an individual's emotional state in conversations, is of great importance in many application scenarios. We organize SemEval-2024 Task 3, named Multimodal Emotion Cause Analysis in Conversations, which aims at extracting all pairs of emotions and their corresponding causes from conversations. Under different modality settings, it consists of two subtasks: Textual Emotion-Cause Pair Extraction in Conversations (TECPE) and Multimodal Emotion-Cause Pair Extraction in Conversations (MECPE). The shared task has attracted 143 registrations and 216 successful submissions. In this paper, we introduce the task, dataset and evaluation settings, summarize the systems of the top teams, and discuss the findings of the participants.

1 Introduction

Understanding emotions is crucial to achieve human-like artificial intelligence, as emotions are intrinsic to humans and significantly influence our cognition, decision-making, and social interactions. Conversation is an important form of human communication and contains a large number of emotions. Furthermore, given that conversation in its natural form is multimodal, many studies have explored multimodal emotion recognition in conversations (ERC), using language, audio and vision modalities (Poria et al., 2019b; Mittal et al., 2020; Lian et al., 2021; Zhao et al., 2022; Zheng et al., 2023).

However, emotion recognition alone is not sufficient to fully understand the intricacies of human emotions. Emotion cause analysis (ECA), the process of identifying the potential causes behind an individual's emotion state, has broad application scenarios such as human-computer interaction, commerce customer service, empathetic conversational agents, and automatic psychotherapy. For example, conversational agents equipped with emotion cause analysis can better understand the user's emotional state, offer empathetic responses, and provide more personalized services. By identifying the cause of the emotional state of a patient, a psychotherapy system can provide more accurate and customized treatments. ECA has gained increasing attention both in academic and practical fields (Ding et al., 2019; Xia et al., 2019; Xia and Ding, 2019; Ding et al., 2020a,b; Poria et al., 2021; Li et al., 2022; An et al., 2023; Wang et al., 2023b). However, to our knowledge, there has not been any evaluation competition conducted specifically for emotion cause analysis in conversations.

To promote research in this direction, we organize a shared task in SemEval-2024, named Multimodal Emotion Cause Analysis in Conversations. Our task consists of two subtasks: Subtask 1 (Textual Emotion-Cause Pair Extraction in Conversations, TECPE) focuses on extracting emotion and textual cause spans solely based on text; Subtask 2 (Multimodal Emotion-Cause Pair Extraction in Conversations, MECPE) involves extracting emotion-cause pairs at the utterance level considering three modalities.

For this shared task, we provide a multimodal emotion cause dataset ECF 2.0 sourced from the sitcom *Friends*. This dataset contains 1,715 conversations and 16,720 utterances, where 12,256 emotion-cause pairs are annotated at the utterance level, covering three modalities (language, audio, and vision). Specifically, in our preliminary work (Wang et al., 2023a), we have constructed a benchmark dataset, Emotion-Cause-in-Friends (ECF 1.0), which contains 1,374 conver-

* Corresponding authors.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 2039–2050

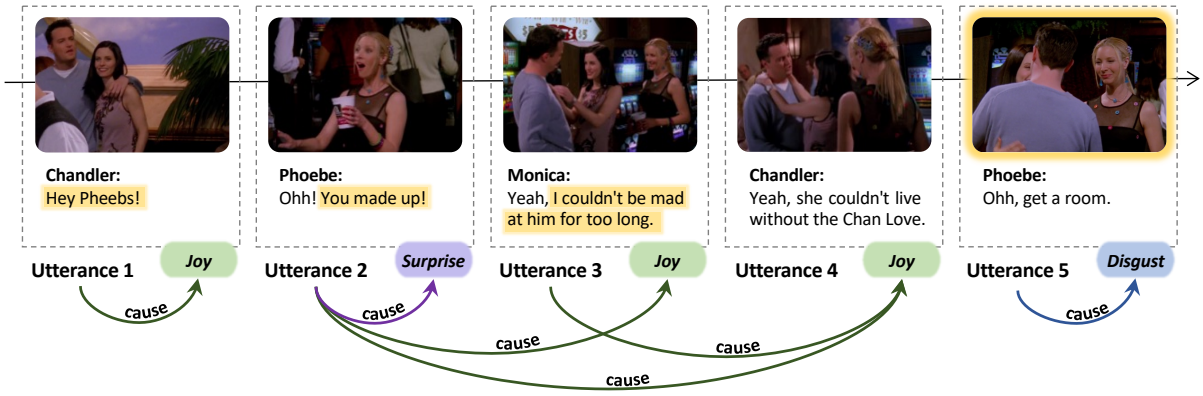


Figure 1: An example of our task and annotated dataset. Each arc points from the cause utterance to the emotion it triggers. The textual cause spans and the visual cause evidence are highlighted in yellow. Background: Chandler and his girlfriend Monica walked into the casino (they had a quarrel earlier but made up soon) and then started a conversation with Phoebe.

sations and 13,619 utterances. On this basis, we have furthermore annotated an extended test set as the evaluation data and provided the span-level annotations of emotion causes within the textual modality.

Our task has attracted 143 registrations and a total of 216 successful submissions during the 16-day evaluation phase. Participants tended to decompose our task into emotion recognition and cause prediction, proposing numerous well-designed pipeline systems. Moreover, many teams applied advanced Large Language Models (LLMs) for emotion cause analysis and achieved promising results. After the evaluation, 18 teams finally submitted system description papers.

2 Task

We clarify the definitions of emotion and cause before introducing the task and dataset. **Emotion** is a psychological state associated with thought, feeling, and behavioral response (Ekman and Davidson, 1994). In computer science, emotions are often described as discrete emotion categories, such as Ekman's six basic emotions, including *Anger*, *Disgust*, *Fear*, *Joy*, *Sadness* and *Surprise* (Ekman, 1971). In conversations, emotions were usually annotated at the utterance level (Li et al., 2017; Hsu et al., 2018; Poria et al., 2019a). **Cause** refers to the objective event or subjective argument that triggers the corresponding emotion (Lee et al., 2010; Russo et al., 2011).

The goal of our shared task, named Multimodal Emotion Cause Analysis in Conversations, is to extract potential pairs of emotions and their corresponding causes from a given conversation. Figure 1 illustrates a typical multimodal conversation scenario, which involves multiple emotions and their corresponding causes. Under different modality settings, we define the following two subtasks:

Subtask 1: Textual Emotion-Cause Pair Extraction in Conversations (TECPE). Extracting all emotion-cause pairs from the given conversation solely based on text, where each pair contains an emotion utterance along with its emotion category and the textual cause span, e.g., (*U3_Joy*, *U2_*"*You made up!*") in Figure 1.

Subtask 2: Multimodal Emotion-Cause Pair Extraction in Conversations (MECPE). It should be noted that sometimes the cause cannot be reflected only in text. As shown in Figure 1, the cause for Phoebe's *Disgust* in *U5* is that Monica and Chandler were kissing in front of her, which is reflected in the visual modality of *U5*. Therefore, we accordingly define this multimodal subtask to extract all emotion-cause pairs in consideration of three modalities (language, audio, and vision). In this subtask, the cause is defined at the utterance level, and each pair contains an emotion utterance along with its emotion category and a cause utterance, e.g., (*U5_Disgust*, *U5*).

3 Dataset

3.1 Data Source

Sitcoms come with real-world-inspired interhuman interactions and usually contain more emotions than other TV series or movies. Based on the famous American sitcom *Friends*, Poria

et al. (2019a) constructed the multimodal conversational dataset MELD by extracting audiovisual clips corresponding to the scripts of the source episodes and annotating each utterance with one of six basic emotions (*Anger*, *Disgust*, *Fear*, *Joy*, *Sadness* and *Surprise*) or *Neutral*. MELD has recently become a widely used benchmark for ERC.

In our preliminary work (Wang et al., 2023a), we chose MELD as the data source and further annotated the causes given emotion annotations, thereby constructing the ECF 1.0 dataset. For this SemEval competition, we release the entire ECF 1.0 dataset as a training set and additionally create a test set as evaluation data, which is also sourced from *Friends*.

3.2 Data Collection

To construct the extended test set, we first crawl the subtitle files of all the episodes of *Friends*, which contains the utterance text and the corresponding timestamps. The subtitles are then separated by scene (scene descriptions are written in square brackets in the subtitle files), and each scene in every episode is viewed as a conversation. If the length of a conversation exceeds 40 utterances, we further divide it into several conversations of random lengths. Conversations included in the ECF 1.0 are removed. Next, we divide the collected conversations into several parts according to their lengths, with each part falling within the length ranges [1, 5], [6, 10], [11, 15], [16, 20], [21, 25], and [26, 35], respectively. Finally, we randomly sample conversations from each part according to the distribution probability of conversation lengths in ECF 1.0, and a total of 400 conversations are sampled for annotation.

3.3 Data Annotation

We employ three graduate students involved in the annotation of the ECF 1.0 dataset to annotate the extended test set. Given a multimodal conversation, they first need to annotate the speaker and emotion category for each utterance, and then further annotate the utterances containing corresponding causes for each non-neutral emotion. If the causes are explicitly expressed in the text, they should also mark the textual cause spans. After annotation, we determine the emotion categories and cause utterances by majority voting, and take the largest boundary (i.e., the union of the spans) as the gold annotation of the textual cause span. If disagreements arise, another expert is invited for the final decision.

Dataset	Modality	Scene	# Ins
Emotion-Stimulus (Ghazi et al., 2015)	T	–	2,414 s
ECE Corpus (Gui et al., 2016)	T	News	2,105 d
NTCIR-13-ECA (Gao et al., 2017)	T	Fiction	2,403 d
Weibo-Emotion (Cheng et al., 2017)	T	Blog	7,000 p
REMAN (Kim and Klinger, 2018)	T	Fiction	1,720 d
GoodNewsEveryone (Bostan et al., 2020)	T	News	5,000 s
RECCON-IE (Poria et al., 2021)	T	Conv	665 u
RECCON-DD (Poria et al., 2021)	T	Conv	11,104 u
ConvECPE (Li et al., 2022)	T,A,V	Conv	7,433 u
ECF 1.0 (Wang et al., 2023a)	T,A,V	Conv	13,619 u
ECF 2.0	T,A,V	Conv	16,720 u

Table 1: Comparison of existing ECA datasets. T, A, and V refer to text, audio, and video. Blog and Conv represent microblog and conversation, and s, d, p and u denote sentence, document, post and utterance.

Items	ECF 1.0	Extended Test	ECF 2.0
Conversations	1,374	341	1,715
Utterances	13,619	3,101	16,720
Emotion (utterances)	7,690	1,821	9,511
Subtask 1 (TECPE)			
Emotion (utterances) with causes	6,761	1,626	8,387
Emotion-cause (span) pairs	9,284	2,256	11,540
Subtask 2 (MECPE)			
Emotion (utterances) with causes	7,081	1,746	8,827
Emotion-cause (utterance) pairs	9,794	2,462	12,256

Table 2: Statistics of our dataset.

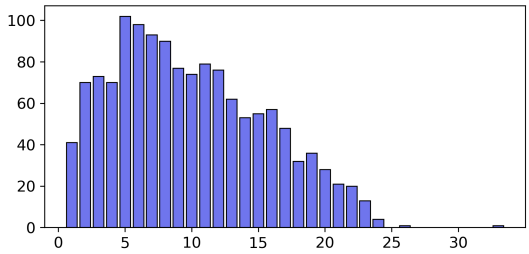
Annotation Cost. The average duration of each conversation in our dataset is 31.6 seconds and it takes about 10 minutes to annotate a conversation. Each annotator would be paid CNY 300 when finishing every 50 conversations, which leads to the basic salary of CNY 36 (USD 5.2) per hour, which is higher than the current average salary in Jiangsu Province, China.

Data Post-processing. We conduct the following post-processing and cleaning of the data:

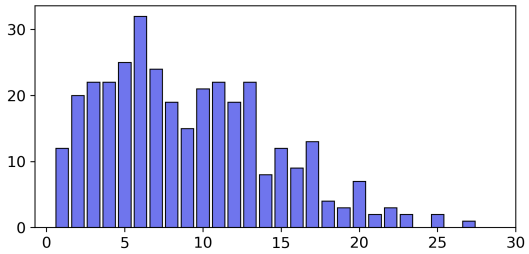
- Correct the utterance text that does not match what the speaker said in the video;
- Correct the timestamps that are not aligned with utterance text;
- Separate the utterance whose segment of timestamps covers two speakers' utterances and modify their timestamps;
- Separate the conversation which spans scenes;
- Discard conversations if there is significant disagreement in annotations and the expert also finds it difficult to determine.

After these steps, we store the text data in JSON files separately for each subtask. For Subtask 2, we use the FFmpeg[1] tool to extract video clips of

[1] https://www.ffmpeg.org

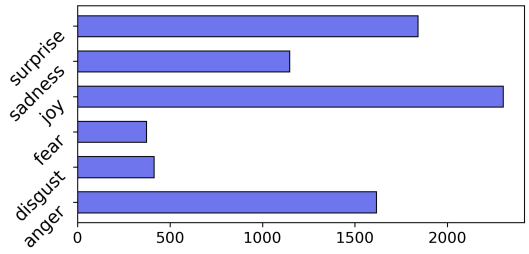


(a) ECF 1.0

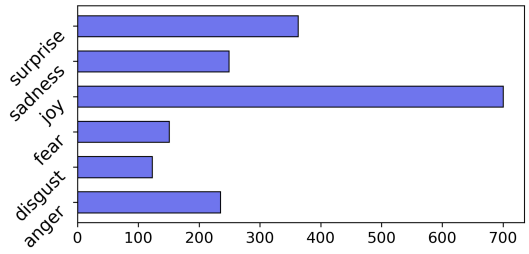


(b) Extended Test set for SemEval-2024

Figure 2: The distribution of conversation lengths. The horizontal axis represents the number of utterances, and the vertical axis represents the number of conversations.



(a) ECF 1.0



(b) Extended Test set for SemEval-2024

Figure 3: The distribution of emotions. The horizontal axis represents the number of utterances, and the vertical axis represents emotion categories.

each utterance from the source episodes based on the start and end timestamps.

3.4 Dataset Statistic

In our preliminary work (Wang et al., 2023a), we have already constructed the ECF 1.0 dataset that contains 1,374 conversations and 13,619 utterances. Furthermore, we have annotated an extended test set specifically for this SemEval evaluation, which together with ECF 1.0 constitutes the **ECF 2.0** dataset[2] that contains 1,715 conversations and 16,720 utterances.

In Table 1, we compare our dataset with the related datasets for ECA, in terms of modality, scene, and size. It is evident that ECF 2.0 is currently the largest available emotion cause dataset.

Table 2 presents the detailed statistics of our dataset for the two subtasks. It can be seen that, in the entire ECF 2.0 dataset, 56.88% of the utterances are labeled with one of the six basic emotions, 92.81% of the emotion utterances have corresponding cause utterances, and 88.18% of the emotion utterances are annotated with textual cause spans.

In addition, as shown in Figure 2 and Figure 3, the newly annotated test set is basically consistent with the original ECF 1.0 dataset in terms of conversation length and emotion distribution.

4 Evaluation

Our SemEval task runs on CodaLab[3]. We released the training data in September 2023, and notified participants to commence model development. The evaluation phase began on January 16, 2024, and ended on January 31, 2024. We mixed the extended test set (consisting of 341 conversations with emotion and cause annotations; the labels are not publicly available) with some noise data (containing 324 conversations, not intended for evaluation) and released them together. Each team is allowed to submit their results up to three times a day.

4.1 Evaluation Metrics

We evaluate the emotion-cause pairs of each emotion category with F_1 scores separately and further calculate a weighted average of F_1 scores across the six emotion categories, denoted as "**w-avg. F_1**". Specifically, for Subtask 1, which involves the textual cause span, we adopt two strategies to determine whether the span is extracted correctly:

- *Strict Match*: A predicted span is regarded as correct if it's the same as one of the annotated spans;

[2]Our dataset is available on Google Drive.

[3]https://codalab.lisn.upsaclay.fr/competitions/16141

Rank	User Name	Team Name	w-avg. S. F_1	w-avg. P. F_1	Main Technologies
1	Mercurialzs	Samsung Research China-Beijing†	0.2300	0.3223	LLaMA2, SpanBERT
2	sachertort	petkaz†	0.1035	0.2640	GPT 3.5, BERT
3	sharadC	UIC NLP GRADS†	0.1839	0.2442	RoBERTa, SpanBERT
4	nicolay-r	nicolay-r†	0.1279	0.2432	Flan-T5
5	Mahshid	AIMA†	0.0218	0.2102	EmoBERTa, DeBERTa
6	jimar	UWBA†	0.0639	0.2084	RoBERTa, BERT
7	Choloe_guo	UIR-ISC†	0.1518	0.1963	BERT, SpanBERT
8	aranjan25	–	0.1431	0.1930	–
9	anaezquerro	LyS†	0.0677	0.1823	BERT
10	wrafal	PWEITINLP†	0.0449	0.0723	GPT-3, SpanBERT
11	ericcui	(˘ɜ˘)-GPT	0.0033	0.0339	–
12	conner	–	0.0000	0.0063	–
13	hpiotr6	–	0.0000	0.0046	–
14	deliagrigorita	–	0.0005	0.0024	–
15	jpcf12	VerbaNexAI Lab†	0.0000	0.0000	Logistic Regression, SpaCy

Table 3: The leaderboard for Subtask 1 (TECPE). "†" indicates that the team has submitted a system description paper to SemEval-2024.

Rank	User Name	Team Name	w-avg. F_1	Modality	Main Technologies
1	Mercurialzs	Samsung Research China-Beijing†	0.3774	T,A,V	LLaMA2, RoBERTa, LLaVA
2	ZhanG_XD	NUS-Emo†	0.3460	T,V	ChatGLM3
3	SZTU-MIPS	SZTU-MIPS†	0.3435	T,A,V	MiniGPT-v2
4	arefa	JMI†	0.2758	T,V	GPT-4V, GPT-3.5
5	Mahshid	AIMA†	0.2584	T	EmoBERTa
6	jimar	UWBA†	0.2506	T,A,V	RoBERTa, BERT
7	julia-bel	DeepPavlov†	0.2057	T,A,V	Video-LLaMA
8	akshettrj	LastResort†	0.1836	T	BiLSTM, CRF
9	oliver_wang	QFNU_CS†	0.1786	T,A,V	BERT
10	MSurfer20	–	0.1708	–	–
11	ayushg2000	–	0.1635	–	–
12	Hidetsune	Hidetsune†	0.1288	T	SpaCy, BERT
13	DuyguA	D-NLP	0.0521	–	–
14	bbgame605065444	NCL†	0.0146	T,A,V	MLP
15	joshuashunk	–	0.0008	–	–

Table 4: The leaderboard for Subtask 2 (MECPE). "†" indicates that the team has submitted a system description paper to SemEval-2024.

- *Proportional Match*: Calculate the overlap proportion of the predicted span and the annotated one.

The evaluation metrics for the two strategies are "w-avg. S. F_1" and "w-avg. P. F_1", respectively. Taking into account the complexity of Subtask 1, we choose "**w-avg. P. F_1**" as the main metric[4] for the ranking.

4.2 Baselines

As mentioned in our previous work (Wang et al., 2023a), for Subtask 2 we also employed the BiLSTM-based ECPE-2steps model as our baseline system. Specifically, we maintain the validation set of the ECF 1.0 datset unchanged and merge the test set into the training set to train the model. The evaluation of the model predictions on the extended test set achieves a weighted average F_1 of **0.1926**.

For Subtask 1, based on the same model, we just convert the cause extraction module in Step 1 from the cause utterance prediction to the prediction of the start index and end index within the utterance, then simply match the indexes as candidate cause spans, followed by emotion-cause pairing and filtering in Step 2. The evaluation result for the weighted average proportional F_1 on the extended test set is **0.1801**.

4.3 Participating Systems and Results

Our competition was created on Codalab in November 2023, and has attracted 143 registrations and a total of 216 submissions. After the evaluation, 18 teams have submitted system de-

[4]Specific calculation details can be found on GitHub.

scription papers.

Team *Samsung Research China-Beijing* (Zhang et al., 2024) won first place in both subtasks, holding a significant lead over the second-place team. Teams *petkaz* (Kazakov et al., 2024) and *UIC NLP GRADS* (Chandakacherla et al., 2024) respectively captured the second and third places in Subtask 1. Teams *NUS-Emo* (Luo et al., 2024) and *SZTU-MIPS* (Cheng et al., 2024) attained second and third positions in Subtask 2. The official leaderboards for Subtask 1 and Subtask 2 are shown in Table 3 and Table 4, respectively.

4.3.1 System Overview

Almost all systems have implemented our task through a two-step framework, first performing the ERC task and then predicting the causes based on emotions. In the following, we briefly introduce the systems from the top teams and some other notable approaches.

Team *Samsung Research China-Beijing* (Zhang et al., 2024) achieved first place in both subtasks with a pipeline framework. They fine-tuned the LLaMA2-based InstructERC (Lei et al., 2023) to extract the emotion category of each utterance in a conversation. For further data augmentation, they added three additional auxiliary tasks based on the original training data strategy of InstructERC. Then, the MuTEC (Bhat and Modi, 2023) and TSAM (Zhang et al., 2022) models are used, respectively, to extract cause spans for Subtask 1 and cause utterances for Subtask 2. They also obtained different multimodal representations through openSMILE (Eyben et al., 2010), LLaVA (Liu et al., 2024), and a self-designed face module to explore the integration of audio-visual information. It should be noted that they used various models for ensemble learning to determine the final prediction.

Team *petkaz* (Kazakov et al., 2024) ranked second in Subtask 1. They fine-tuned GPT 3.5 (Ouyang et al., 2022) for emotion classification and then used a BiLSTM-based neural network to detect cause utterances. The cause extractor model is initialized with BERT (Devlin et al., 2019), followed by three BiLSTM layers. They treat the entire cause utterance as a cause span.

Team *NUS-Emo* (Luo et al., 2024) achieved the second highest score in Subtask 2. First, they conducted zero-shot testing experiments to evaluate multiple LLMs, including OPT-IML3 (Iyer et al., 2022), Instruct-GPT4 (Peng et al., 2023), Flan-T5 (Chung et al., 2022), and ChatGLM (Du et al., 2022). ChatGLM3-6B is ultimately selected as its backbone model based on its superior performance. They designed an emotion-cause-aware instruction-tuning mechanism to update the LLM and incorporated the visual representation from the ImageBind (Girdhar et al., 2023) encoder.

Team *UIC NLP GRADS* (Chandakacherla et al., 2024) achieved the third place in Subtask 1, and their system performed well in the strict metric, ranking second. They fine-tuned RoBERTa (Liu et al., 2019) for emotion classification, and then further fine-tuned a SpanBERT (Joshi et al., 2019) model that had been fine-tuned in SQuAD 2.0 (Rajpurkar et al., 2018), to predict cause spans in QA format.

Team *SZTU-MIPS* (Cheng et al., 2024) ranked third in Subtask 2. They integrated text, audio, and image modalities for emotion recognition and adopted the MiniGPTv2 model (Chen et al., 2023) for multimodal cause extraction. Specifically, textual features are obtained from InstructERC, while acoustic features are extracted using HuBERT (Hsu et al., 2021). For visual modality, faces are first extracted using OpenFace (Baltrusaitis et al., 2016) from video frames, followed by extraction of facial features using expMAE (Cheng et al., 2023).

Team *nicolay-r* (Rusnachenko and Liang, 2024) finetuned Flan-T5 by designing the chain of thoughts for emotion causes based on the Three-Hop Reasoning (THOR) framework (Fei et al., 2023), to predict the emotion of the current utterance and the emotion caused by the current utterance towards the target utterance. Their reasoning revision methodology and rule-based span correction technique bring further improvements.

Team *JMI* (Arefa et al., 2024) implemented two different approaches. In their best system, they used in-context learning using GPT 3.5 for emotion prediction and cause prediction, respectively. Conversation-level video descriptions were extracted via GPT-4V (Yang et al., 2023) to provide more context to GPT 3.5. In addition, they also fine-tuned two separate Llama2 (Touvron et al., 2023) models to recognize emotions and extract causes.

Team *AIMA* (Abootorabi et al., 2024) fine-tuned EmoBERTa (Kim and Vossen, 2021) for emotion classification and then obtained the emotion-cause pairs via a Transformer-based encoder. After finding the pairs, they further fine-tuned the

DeBERTa (He et al., 2021) that had been fine-tuned on SQuAD 2.0 to extract the cause spans for Subtask 1.

Team *UWBA* (Baloun et al., 2024) fused the features of three modalities at the utterance level and then used them for emotion classification and pair prediction. It is interesting that they summarized five span categories (*Whole Utterance*, *First part*, *Last part*, *Middle part*, *Other*) through observations of training data, and then trained a classifier to further predict textual cause spans in cause utterance.

Furthermore, Team *DeepPavlov* (Belikova and Kosenko, 2024) investigated the performance of Video-LLaMA (Zhang et al., 2023) in several modes and found that model fine-tuning yields notable improvements in emotion and cause classification. Team *PWEITINLP* (Levchenko et al., 2024) utilized GPT-3 for emotion classification. Some other Teams, including *UIR-ISC* (Guo et al., 2024), *LyS* (Ezquerro and Vilares, 2024), *QFNU_CS* (Wang et al., 2024) and *Hidetsune* (Takahashi, 2024), all employed BERT-based models to address our task, among which *LyS* proposed an end-to-end model comprising a BERT encoder and a graph-based decoder to identify emotion cause relations. Team *LastResort* (Mathur et al., 2024) tackled our task as sequence labeling problems and used BiLSTM followed by a CRF layer to solve it. Team *NCL* (Li et al., 2024) solely utilized pre-trained models to extract features from three modalities. Team *VerbaNexAI Lab* (Garcia et al., 2024) demonstrated the inadequacy of machine learning techniques alone for emotion cause analysis.

4.3.2 Discussion

Our task, Multimodal Emotion Cause Analysis in Conversations, involves informal real-life conversations and complex audio-visual scenes. Additionally, emotions exhibit strong subjectivity, and we have observed that even humans sometimes struggle to accurately identify emotions and their causes. This complexity underscores the intricate nature of human emotions and the nuanced contexts in which they occur, posing a substantial challenge for data annotation and subsequent model development.

Dataset Bias. Emotion category imbalance is an inherent problem in the ERC task (Li et al., 2017; Hsu et al., 2018; Poria et al., 2019a), aligning with real-world phenomena where people tend to express positive emotions like *joy* more frequently in their daily communications, while expressions of *disgust* and *fear* are less common. Our dataset is sourced from TV series that closely resemble the real world, naturally also exhibiting an imbalance in emotions, as illustrated in Figure 3. However, such an imbalance may adversely affect a model's ability to learn and generalize across different emotions, potentially leading to biases towards frequently expressed emotions (Kazakov et al., 2024; Chandakacherla et al., 2024). Moreover, emotion cause datasets often have a noticeable pattern in the location of causes and emotions. Some systems rely on this position bias, either by using a fixed window size or by direct post-processing to add the emotion utterance as the cause (Rusnachenko and Liang, 2024; Arefa et al., 2024), which overlooks the effective semantic connections between distant contexts and may lead to poor generalization capabilities for unseen data where the cause is not in proximity to the emotion. In the future, LLMs can be leveraged to assist with annotation to expand the diversity of datasets available for fine-tuning, which encompass a wider range of emotional expressions and cultural backgrounds. This can mitigate existing dataset biases and enhance the model's applicability and generalizability across various scenarios.

Utilization of LLMs. Recently, LLMs have exhibited remarkable capabilities in a wide range of tasks and are rapidly advancing the field of natural language processing. Therefore, LLMs are allowed to be used in our competition. It is evident that about a third of the teams have used LLMs for emotion cause analysis, and most of them are ranked at the top. However, some participants have observed that LLMs perform poorly in zero-shot and few-shot settings on emotion and cause recognition tasks (Kazakov et al., 2024; Arefa et al., 2024; Belikova and Kosenko, 2024), indicating a crucial need for task-specific fine-tuning. Furthermore, prompt engineering is essential, as LLMs often produce hallucinations or unstructured outputs. Due to resource and cost constraints, most researchers cannot take full advantage of the strongest capabilities of LLM. Future research is encouraged to explore ways to enhance lightweight models or to bridge the gap between pre-training and downstream tasks, thereby augmenting LLMs' ability to understand emotions.

Potential of Multimodal Information. Multimodal information is important for discovering both emotions and their causes in conversations. In our daily communications, we depend not only on the speaker's voice intonation and facial expressions to perceive his emotions, but also on some auditory and visual scenes to speculate the potential causes that trigger the emotions of speakers beyond text. However, some participants found that the introduction of audio or visual modalities results in minimal improvements or even a decrease in system performance (Zhang et al., 2024; Cheng et al., 2024; Baloun et al., 2024). This issue arises partly due to the characteristics of our dataset, which involves a large number of complex visual scenes but few visual cause clues, leading to the introduction of noise. Another limiting factor might be that multimodal feature extraction methods are not advanced enough or fusion strategies are not effective enough. The challenges that require further exploration include the effective interaction and fusion of multimodal information, as well as the perception, understanding, and utilization of audiovisual scenes. Furthermore, there is a demand for more high-quality data sets on multimodal emotion cause analysis to support research in this area.

5 Conclusions

In this paper, we describe the SemEval-2024 Task 3 named Multimodal Emotion Cause Analysis in Conversations, which aims to extract all potential pairs of emotions and their corresponding causes from a conversation. The shared task has attracted 143 registrations and 216 successful submissions. We provide detailed descriptions of task definition and data annotation, summarize participating systems, and discuss their findings.

As an important direction of affective computing, multimodal emotion cause analysis in conversation plays an important role in many real-world applications. We hope that our research and resources can contribute towards the design of future systems in this direction.

6 Ethics Statement

Our **ECF 2.0** dataset is annotated on the basis of the MELD dataset [5] which is licensed under the GNU General Public License v3.0 and is used only for scientific research. We do not share personal information and do not release sensitive content that can be harmful to any individual or community. Conducting multimodal emotion cause analysis will help us better understand emotions in human conversations, build human-machine dialogue systems, and contribute to society and human well-being.

[5] https://github.com/declare-lab/MELD

Acknowledgements

We express our sincere gratitude to the annotators who contributed to constructing the dataset, laying a solid foundation for our research. We also extend our heartfelt gratitude to all the participants who participated in our competition, especially the teams who submitted system description papers and completed the reviewing tasks assigned to them. Furthermore, we thank the anonymous reviewers for their invaluable feedback and insightful comments.

References

Mohammad Mahdi Abootorabi, Nona Ghazizadeh, Seyed Arshan Dalili, Alireza Ghahramani Kure, Mahshid Dehghani, and Ehsaneddin Asgari. 2024. Aima at semeval-2024 task 10: History-based emotion recognition in hindi-english code-mixed conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1714–1720, Mexico City, Mexico. Association for Computational Linguistics.

Jiaming An, Zixiang Ding, Ke Li, and Rui Xia. 2023. Global-view and speaker-aware emotion cause extraction in conversations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Arefa, Mohammed Abbas Ansari, Chandni Saxena, and TANVIR AHMAD. 2024. Jmi at semeval 2024 task 3: Two-step approach for multimodal ecac using in-context learning with gpt and instruction-tuned llama models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1571–1586, Mexico City, Mexico. Association for Computational Linguistics.

Josef Baloun, Jiri Martinek, Ladislav Lenc, Pavel Kral, Matěj Zeman, and Lukáš Vlček. 2024. Uwba at semeval-2024 task 3: Dialogue representation and multimodal fusion for emotion cause analysis. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 309–318, Mexico City, Mexico. Association for Computational Linguistics.

Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. 2016. Openface: An open source facial behavior analysis toolkit. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10.

Julia Belikova and Dmitrii Kosenko. 2024. Deep-pavlov at semeval-2024 task 3: Multimodal large language models in emotion reasoning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1757–1767, Mexico City, Mexico. Association for Computational Linguistics.

Ashwani Bhat and Ashutosh Modi. 2023. Multi-task learning framework for extracting emotion cause span and entailment in conversations. In *Transfer Learning for Natural Language Processing Workshop*, pages 33–51. PMLR.

Laura Ana Maria Bostan, Evgeny Kim, and Roman Klinger. 2020. Goodnewseveryone: A corpus of news headlines annotated with emotions, semantic roles, and reader perception. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1554–1566.

Sharad Chandakacherla, Vaibhav Bhargava, and Natalie Parde. 2024. Uic nlp grads at semeval-2024 task 3: Two-step disjoint modeling for emotion-cause pair extraction. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1362–1368, Mexico City, Mexico. Association for Computational Linguistics.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *ArXiv*, abs/2310.09478.

Xiyao Cheng, Ying Chen, Bixiao Cheng, Shoushan Li, and Guodong Zhou. 2017. An emotion cause corpus for chinese microblogs with multiple-user structures. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):1–19.

Zebang Cheng, Yuxiang Lin, Zhaoru Chen, Xiang Li, Shuyi Mao, Fan Zhang, Daijun Ding, Bowen Zhang, and Xiaojiang Peng. 2023. Semi-supervised multimodal emotion recognition with expression mae. *Proceedings of the 31st ACM International Conference on Multimedia*.

Zebang Cheng, Fuqiang Niu, Yuxiang Lin, Zhi-Qi Cheng, Xiaojiang Peng, and Bowen Zhang. 2024. Mips at semeval-2024 task 3: Multimodal emotion-cause pair extraction in conversations with multimodal language models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 653–660, Mexico City, Mexico. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Zixiang Ding, Huihui He, Mengran Zhang, and Rui Xia. 2019. From independent prediction to reordered prediction: Integrating relative position and global label information to emotion cause identification. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 6343–6350.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020a. ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *Association for Computational Linguistics (ACL)*, pages 3161–3170.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020b. End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3574–3583.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

P. Ekman. 1971. Universals and cultural differences in facial expressions of emotion. *Nebraska Symposium on Motivation. Nebraska Symposium on Motivation*, Vol. 19.

Paul Ed Ekman and Richard J Davidson. 1994. *The nature of emotion: Fundamental questions.* Oxford University Press.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462.

Ana Ezquerro and David Vilares. 2024. Lys at semeval-2024 task 3: An early prototype for end-to-end multimodal emotion linking as graph-based parsing. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1254–1261, Mexico City, Mexico. Association for Computational Linguistics.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat seng Chua. 2023. Reasoning implicit sentiment

with chain-of-thought prompting. In *Annual Meeting of the Association for Computational Linguistics*.

Qinghong Gao, Jiannan Hu, Ruifeng Xu, Gui Lin, Yulan He, Qin Lu, and Kam-Fai Wong. 2017. Overview of ntcir-13 eca task. In *Proceedings of the NTCIR-13 Conference*.

Santiago Garcia, Elizabeth Martinez, Juan Cuadrado, Juan Carlos Martinez-Santos, and Edwin Puertas. 2024. Verbanexai lab at semeval-2024 task 10: Emotion recognition and reasoning in mixed-coded conversations based on an nrc vad approach. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1321–1327, Mexico City, Mexico. Association for Computational Linguistics.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 152–165. Springer.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190.

Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, Yu Zhou, et al. 2016. Event-driven emotion cause extraction with corpus construction. In *EMNLP*, pages 1639–1649. World Scientific.

Hongyu Guo, Xueyao Zhang, Yiyang Chen, Lin Deng, and Binyang Li. 2024. Uir-isc at semeval-2024 task 3: Textual emotion-cause pair extraction in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 757–763, Mexico City, Mexico. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Chao-Chun Hsu, Sheng-Yeh Chen, Chuan-Chun Kuo, Ting-Hao Huang, and Lun-Wei Ku. 2018. Emotionlines: An emotion corpus of multi-party conversations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdel rahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Srinivas Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Veselin Stoyanov. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *ArXiv*, abs/2212.12017.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Roman Kazakov, Kseniia Petukhova, and Ekaterina Kochmar. 2024. Petkaz at semeval-2024 task 3: Advancing emotion classification with an llm for emotion-cause pair extraction in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1116–1123, Mexico City, Mexico. Association for Computational Linguistics.

Evgeny Kim and Roman Klinger. 2018. Who feels what and why? annotation of a literature corpus with semantic roles of emotions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1345–1359.

Taewoon Kim and Piek Vossen. 2021. Emoberta: Speaker-aware emotion recognition in conversation with roberta. *ArXiv*, abs/2108.12009.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *NAACL HLT Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 45–53.

Shanglin Lei, Guanting Dong, Xiaoping Wang, Keheng Wang, and Sirui Wang. 2023. Instructerc: Reforming emotion recognition in conversation with a retrieval multi-task llms framework. *arXiv preprint arXiv:2309.11911*.

Sofiia Levchenko, Rafał Wolert, and Piotr Andruszkiewicz. 2024. Pweitinlp at semeval-2024 task 3: Two step emotion cause analysis. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1086–1094, Mexico City, Mexico. Association for Computational Linguistics.

Shu Li, Zicen Liao, and Huizhi Liang. 2024. Ncl team at semeval-2024 task 3: Fusing multimodal pretraining embeddings for emotion cause prediction in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 285–290, Mexico City, Mexico. Association for Computational Linguistics.

Wei Li, Yang Li, Vlad Pandelea, Mengshi Ge, Luyao Zhu, and Erik Cambria. 2022. Ecpec: emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Zheng Lian, Bin Liu, and Jianhua Tao. 2021. Ctnet: Conversational transformer network for emotion recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:985–1000.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Meng Luo, Han Zhang, Shengqiong Wu, Bobo Li, Hong Han, and Hao Fei. 2024. Nus-emo at semeval-2024 task 3: Instruction-tuning llm for multimodal emotion-cause analysis in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1599–1606, Mexico City, Mexico. Association for Computational Linguistics.

Suyash Vardhan Mathur, Akshett Jindal, Hardik Mittal, and Manish Shrivastava. 2024. Lastresort at semeval-2024 task 3: Exploring multimodal emotion cause pair extraction as sequence labelling task. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1194–1201, Mexico City, Mexico. Association for Computational Linguistics.

Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. 2020. M3er: Multiplicative multimodal emotion recognition using facial, textual, and speech cues. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1359–1367.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *ArXiv*, abs/2304.03277.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019a. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Pengfei Hong, Romila Ghosh, Abhinaba Roy, Niyati Chhaya, et al. 2021. Recognizing emotion cause in conversations. *Cognitive Computation*, pages 1–16.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019b. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *ArXiv*, abs/1806.03822.

Nicolay Rusnachenko and Huizhi Liang. 2024. nicolay-r at semeval-2024 task 3: Using flan-t5 for reasoning emotion cause in conversations with chain-of-thought on emotion states. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 22–27, Mexico City, Mexico. Association for Computational Linguistics.

Irene Russo, Tommaso Caselli, Francesco Rubino, Ester Boldrini, and Patricio Martínez-Barco. 2011. Emocause: an easy-adaptable approach to emotion cause contexts. In *Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 153–160.

Hidetsune Takahashi. 2024. Hidetsune at semeval-2024 task 3: A simple textual approach to emotion classification and emotion cause analysis in conversations using machine learning and next sentence prediction. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 354–357, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu,

Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2023a. Multimodal emotion-cause pair extraction in conversations. *IEEE Transactions on Affective Computing*, 14(3):1832–1844.

Fanfan Wang, Jianfei Yu, and Rui Xia. 2023b. Generative emotion cause triplet extraction in conversations with commonsense knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3952–3963.

Zining Wang, Yanchao Zhao, Guanghui Han, and Yang Song. 2024. Qfnu_cs at semeval-2024 task 3: A hybrid pre-trained model based approach for multimodal emotion-cause pair extraction task. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 349–353, Mexico City, Mexico. Association for Computational Linguistics.

Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012.

Rui Xia, Mengran Zhang, and Zixiang Ding. 2019. RTHN: A RNN-transformer hierarchical network for emotion cause extraction. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5285–5291.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1.

Duzhen Zhang, Zhen Yang, Fandong Meng, Xiuyi Chen, and Jie Zhou. 2022. Tsam: A two-stream attention model for causal emotion entailment. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6762–6772.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *ArXiv*, abs/2306.02858.

Shen Zhang, Haojie Zhang, Jing Zhang, Xudong Zhang, Yimeng Zhuang, and Jinting Wu. 2024. Samsung research china-beijing at semeval-2024 task 3: A multi-stage framework for emotion-cause pair extraction in conversations. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 536–546, Mexico City, Mexico. Association for Computational Linguistics.

Jinming Zhao, Tenggan Zhang, Jingwen Hu, Yuchen Liu, Qin Jin, Xinchao Wang, and Haizhou Li. 2022. M3ed: Multi-modal multi-scene multi-label emotional dialogue database. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5699–5710.

Wenjie Zheng, Jianfei Yu, Rui Xia, and Shijin Wang. 2023. A facial expression-aware multimodal multi-task learning framework for emotion recognition in multi-party conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15445–15459.

SheffieldVeraAI at SemEval-2024 Task 4: Prompting and fine-tuning a Large Vision-Language Model for Binary Classification of Persuasion Techniques in Memes

Charlie Grimshaw, Kalina Bontcheva and **Xingyi Song**
Department of Computer Science
University of Sheffield
Sheffield, UK
{cgrimshaw1, k.bontcheva, x.song}@sheffield.ac.uk

Abstract

This paper describes our approach for SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. Specifically, we concentrate on Subtask 2b, a binary classification challenge that entails categorizing memes as either "propagandistic" or "non-propagandistic". To address this task, we utilized the large multimodal pretrained model, LLaVa. We explored various prompting strategies and fine-tuning methods, and observed that the model, when not fine-tuned but provided with a few-shot learning examples, achieved the best performance. Additionally, we enhanced the model's multilingual capabilities by integrating a machine translation model. Our system secured the 2nd place in the Arabic language category.

1 Introduction

Research of online misinformation is growing (Chaudhari and Pawar, 2021) as fake news and propagandistic content spreads further and replaces more real news on social media, detrimentally impacting society, including loss of lives, loss of health and economic loss (Muhammed T and Mathew, 2022). A common online propaganda format is a meme, where text and image(s) are combined to share a message, often political (Guo et al., 2020). This paper describes SheffieldVeraAI's approach for SemEval 2024 Task 4 Subtask 2b, involving detecting the presence of persuasion technique(s) within memes, a binary visual/textual classification task (Dimitrov et al., 2024). The previous research on this task including (Feng et al., 2021) (Tian et al., 2021) (Li et al., 2021) used non-autoregressive encoder representation techniques, using models such as BERT (Devlin et al., 2018) or RoBERTa (Liu et al., 2019), fused with a vision representation model such as ResNet (He et al., 2015).

Unlike the previous research, we experimented with a different approach to use and train an auto-regressive vision-language model that receives image and text as input, outputting text only. Prompting the model with a meme and expecting the model to generate a classification output. Specifically, we use the LLaVa-1.5 model (Liu et al., 2023a), which directly projects an image encoding into tokens computed as text tokens by an LLM. This technique allows us to utilise the LLMs' "knowledge" of persuasion techniques they have learnt through massive pre-training and explained outputs through prompting, improving the model's interoperability and error analysis.

1.1 Contributions

- Show that a pre-trained auto-regressive large visual language model can be prompted for binary persuasion classification.
- Show that prompting with translated text is a viable method, achieving 2nd place in the Arabic leaderboard, using an English-only model.

2 Background

Previous SemEval tasks have looked at this problem of online misinformation/persuasion/propaganda:

- **SemEval 2020 Task 11 - "Detection of Propaganda Techniques in News Articles" (Martino et al., 2020).** This task involved span and technique investigation on text-only news articles.
- **SemEval 2021 Task 6 - "Detection of Persuasion Techniques in Texts and Images" (Dimitrov et al., 2021).** The first task relating to persuasion techniques involved a subtask with images, which required classifying propaganda techniques within memes.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 2051–2056

- **SemEval 2023 Task 3 - "Detecting the Genre, the Framing, and the Persuasion Techniques in Online News in a Multilingual Setup" (Piskorski et al., 2023).** This task is similar to SemEval 2020 Task 11, which contains no visual content or news articles while adding genre and framing detection.

This theme of persuasion technique detection is prominent in recent years of SemEval, with the most similar task being SemEval 2021 Task 6 Subtask 3, which involved visual and textual persuasion techniques in memes.

2.1 Task Description

In this work, our focus is on Subtask 2b, which aims to determine whether at least one persuasion technique is present in the meme or no technique is present. This task provides both the original image and the text transcriptions. The detailed data structure is outlined as follows:

- unique *id* of the sample. e.g. 12345
- The image of the meme, an example can be found in Figure 1
- A transcription of text within the image content of the meme. For example: ‘‘GIVE A THUMBS UP IF YOU\\nSTILL SUPPORT TRUMP\\n”
- A label which is either **propagandisitic** or **non-propagandistic**. A meme is propagandistic if it contains one or more of the 22 persuasion techniques defined by the task organisers.

The language of the meme and transcription is either English, Bulgarian, North Macedonian or Arabic. The language of the meme and transcription always match.

3 System Overview

Our system follows these steps:

1. Fine-tune LLaVa with LoRA (Hu et al., 2021) using pre-processed English training data. (Optional; our final system is untrained).
2. Translate Bulgarian, North Macedonian and Arabic transcriptions to English using NLLB (Team et al., 2022).

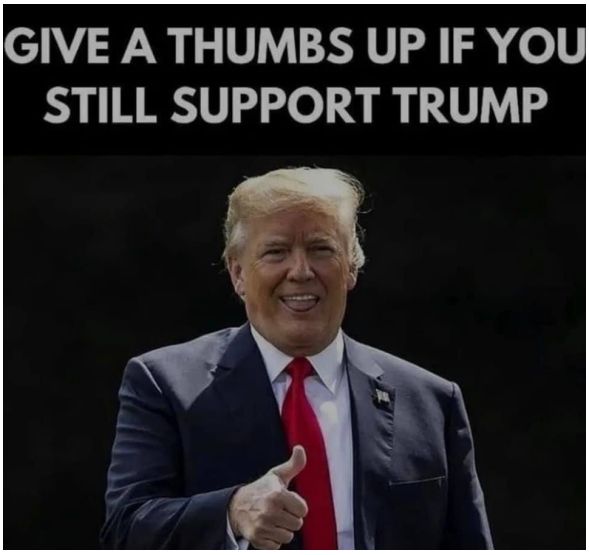


Figure 1: Example of a propagandistic image from the task

3. Prompt LLaVa for binary classification of persuasion techniques, giving a few-shot example.

3.1 LLaVa

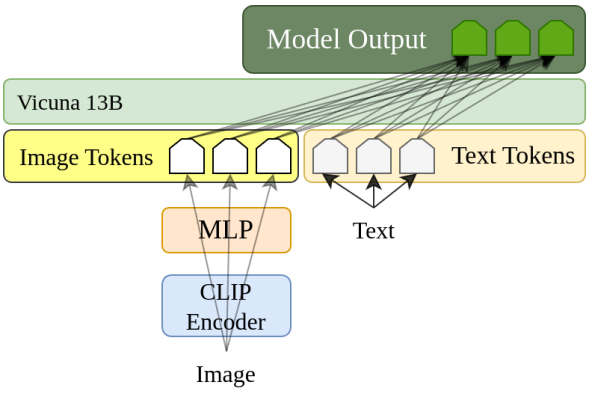


Figure 2: Diagram of LLaVa 1.5 architecture, modelled from original paper (Liu et al., 2023b).

The model we use for this task is called LLaVa (Large Language and Vision Assistant) (Liu et al., 2023b) (Liu et al., 2023a). We are using the 13B parameter version of LLaVa-1.5. We use the original author’s public code, available on GitHub[1], for training and inference. LLaVa is an English-only end-to-end fine-tuned Large Vision-Language Model (LVLM) trained on Chat-GPT4 (OpenAI, 2023) generated instruction-following data. It handles image inputs using a trained projection, a multi-layer perceptron (MLP) in LLaVa-

[1]https://github.com/haotian-liu/LLaVA

1.5, that projects image features from a CLIP encoder (Ramesh et al., 2022) into the word embedding space of an LLM, Vicuna-13B (Chiang et al., 2023). We experiment with prompting LLaVa for binary classification with a meme image and other information that could improve the models' performance and fine-tune the model using the training set for the task. We fine-tune the model using LoRA, a widely used training technique for reducing the number of trainable parameters. We use LoRA to reduce training time and required GPU memory.

3.2 Machine Translation

For the three unseen languages (Bulgarian, North Macedonian and Bulgarian), that are part of the test set, we use the machine translation model NLLB, as this model can translate English into all three unseen languages and is trained at sentence-level which matches the short form text within memes. As LLaVa is English only, we will translate all non-English transcriptions from the test set into English and use these as inputs for LLaVa, allowing LLaVa still to see the visual content of the original meme while receiving text input it understands.

4 Experimental Setup

4.1 Data processing

The dataset provided was split into 3 sets: 1200 training, 150 validation and 300 unlabeled development examples used for early testing and a leaderboard available before the test set was released. These splits were entirely in English. The final test set contained 600 memes in English, 100 in Bulgarian, 100 in North Macedonian and 160 in Arabic. We used the training set for fine-tuning our model, the validation set for finding the best prompts for our model, and the development set to get our results when the labels were released.

To preprocess the data, we removed all new lines and non-Latin characters from English and translated all non-English text from the test set into English before inputting them into the model.

4.2 Hyperparameters

The two hyperparameters we experimented with were the LoRA parameters rank (r) and α. r controls the trainable parameters for fine-tuning, and α is a scaling parameter that affects how much the LoRA adaption weights affect the base model weights. We experimented with every combination of the following values

- r - [8, 16]
- α - [4, 8, 16, 32]

We did experiment with numbers outside this range, but they only worsened the model's performance. We trained the model for 1 epoch, using a single 80GB A100 GPU. We used Python 3.10.13 and the Hugging Face models *liuhaotian/llava-v1.5-13b*[2] and *facebook/nllb-200-3.3B*[3].

4.3 Prompting

We experimented with different prompting techniques. We report the results in Table 1. We tested each technique using the development set as follows:

- **Basic Prompt:**

  USER: <image>\n
  Does this meme contain any propagandistic or persuasive techniques? Answer with "yes" or "no"\n
  ASSISTANT:

- **Meme Text Included:**

  USER: <image>\n
  This meme contains the text: <text>.
  Does this meme contain any propagandistic or persuasive techniques? Answer with "yes" or "no"\n
  ASSISTANT:

- **Persuasive/Propaganda:**
  Here, we experimented with using different words for the techniques.

  USER: <image>\n
  This meme contains the text: <text>.
  Does this meme contain any <propaganda/persuasive> techniques? Answer with "yes" or "no"\n
  ASSISTANT:

- **Examples of Persuasion techniques:**
  Here we experiment by providing an example of some persuasion techniques. We tested every combination of 1-5 persuasion techniques from subtask 2b and found the following prompt to perform the best.

[2]https://huggingface.co/liuhaotian/llava-v1.5-13b
[3]https://huggingface.co/facebook/nllb-200-3.3B

USER: <image>\n
You are tasked with detecting the presence of propaganda techniques in memes. Examples of propaganda techniques are: [Black-and-white Fallacy/Dictatorship, Doubt, Slogans, Appeal to authority, Bandwagon] This meme contains the text: <text>. Does this meme contain any propaganda techniques? Answer with just "Yes" or "No" \n
ASSISTANT:

- **Few-shot example prompt:** We experimented with providing an example of a propagandistic meme within the prompt, hoping to improve the model's classification performance. We could only give the model the transcription from a propagandistic meme, as the LLaVa model was only trained to receive one input image.

  USER: <image>\n
  You are tasked with detecting the presence of propaganda techniques in memes. Some but not all examples of propaganda techniques are: [Black-and-white Fallacy/Dictatorship, Doubt, Slogans, Appeal to authority, Bandwagon]. For example, a meme that contains the text: [American democracy and the Soviet system may peacefully exist side by side and compete with each other. But one cannot evolve into the other. (J. Stalin)] contains propaganda techniques. This meme contains the text: [<Meme Transcription>]. Does this meme contain any propaganda techniques? Answer with just "Yes" or "No".\n
  ASSISTANT:

  We use this final prompt when testing and fine-tuning our model. For fine-tuning, we pair it with a desired output of *yes* if the meme is propagandistic and *no* otherwise. Before evaluating the models, we convert *yes* and *no* back to their corresponding labels.

5 Results

Table 2 presents the results of fine-tuning our model on training data and testing it on the development set. As reported, fine-tuning using our experimental setup only worsened the model's performance, so we chose the untrained baseline LLaVa-1.5 model for submission using the test set.

Table 3 presents the final results for the leaderboard for our attempt at the task. Our worst performing result was in the English language. However, we performed better in the unseen languages, with our best result being **2nd place in Arabic.**

5.1 Error Analysis

Figure 3 shows a confusion matrix for the untrained model on the development set. We can see that by proportion, it made more errors predicting propagandistic memes as non-propagandistic than otherwise.

By prompting the model, we can "ask" the model why it made a classification for any meme, which increases the explainability and error analysis capabilities of a model like this. We provide output explanations for two examples where it predicts pro-

Technique	Macro-F1
Basic Prompt	0.42
Meme Text Included	0.45
Persuasive	0.50
Propagandistic	0.60
Example techniques	0.65
Few-shot example	**0.66**

Table 1: Results from using different prompting techniques. The best results are marked as **bold**

.

r	α	Macro-F1	Micro-F1
16	32	0.62	0.73
	16	0.60	0.72
	8	0.53	0.70
	4	0.54	0.70
8	32	0.65	0.74
	16	0.61	0.72
	8	0.50	0.70
	4	0.57	0.71
Untrained		**0.69**	**0.75**

Table 2: Results on the dev set from our standard training strategy. Best results are marked as **bold.**

Final Submission		
Language	Macro-F1	Place
English	0.64	15
Bulgarian	0.54	8
North Macedonian	0.46	10
Arabic	0.61	2

Table 3: Subtask 2b results from the final submission, with placement in the leaderboard for each language.

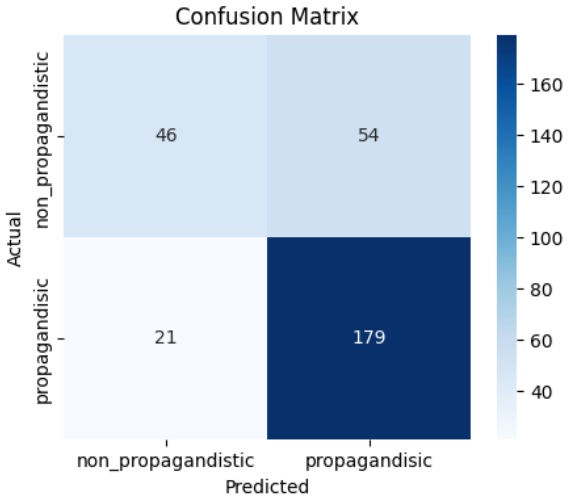


Figure 3: Confusion Matrix of untrained LLaVa on the dev set.

pagandistic (P) as propagandistic (NP) and where it predicts NP as P.

- **Classifying P as NP** - Dev Set ID 28493:

  > "The meme does not contain any of the propaganda techniques listed. It is a simple image of two men shaking hands, with a caption that is not promoting any particular ideology or using any manipulative techniques. The meme is not trying to persuade or influence the viewer's opinion, but rather it is simply presenting a visual representation of a positive interaction between two individuals."

  The original meme requires context about Russian President Vladimir Putin's relationship with intergovernmental forum G20, which the model fails to recognise. It claims the meme doesn't contain any of the propaganda techniques listed. This is likely referring to the example persuasion technique provided in the original prompt, which the model may be confusing for the **only** persuasion techniques to look for.

- **Classifying NP as P** - Dev Set ID 45439:

  > "Yes, this meme contains a propaganda technique called "Slogans." The meme is promoting a political message, specifically advocating for expanding the court. The use of the hashtag WENEEDNINE and the phrase "Expand the Court" are examples of slogans that aim to influence public opinion and support for the political cause. Slogans are often used in propaganda to create a memorable and catchy phrase that can be easily shared and repeated, thus increasing the reach and impact of the message."

  Although the meme does contain slogans in its image, they aren't part of the message and joke of the meme; the slogans are there because the meme is about a politician who happens to be photographed standing behind a slogan. However, the model fails to make this distinction.

6 Conclusion

We have presented our system for SemEval 2024 Task 4 Subtask 2b. We prompted and fine-tuned an auto-regressive large visual language model and showed that LLaVa can be used for non-English persuasion technique detection in memes through improved prompting and machine translation, with our best ranking being 2nd on the Arabic leaderboard. We experimented with different techniques for prompting to discover which produced the best-performing output. We also analysed the type of errors LLaVa can produce when classifying memes, showing how this model can easily be prompted for explainability. Further work is required to improve training LVLMs for this task, as we could not see improvements through fine-tuning.

Acknowledgements

This work has been co-funded by the European Union under the Horizon Europe vera.ai (grant 101070093) and Vigilant (grant 101073921) projects and the UK's innovation agency (Innovate UK) grants 10039055 and 10039039.

References

Deptii Devendra Chaudhari and Ambika Vishal Pawar. 2021. Propaganda analysis in social media: a bibliometric review. *Information Discovery and Delivery*, 49(1):57–70. Publisher: Emerald Publishing Limited.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. SemEval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th international workshop on semantic evaluation*, SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 Task 6: Detection of Persuasion Techniques in Texts and Images. ArXiv:2105.09284 [cs].

Zhida Feng, Jiji Tang, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, and Li Chen. 2021. Alpha at SemEval-2021 task 6: Transformer based propaganda classification. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 99–104, Online. Association for Computational Linguistics.

Bin Guo, Yasan Ding, Lina Yao, Yunji Liang, and Zhiwen Yu. 2020. The future of false information detection on social media: New perspectives and trends. *ACM Comput. Surv.*, 53(4).

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Peiguang Li, Xuan Li, and Xian Sun. 2021. 1213Li at SemEval-2021 task 6: Detection of propaganda with multi-modal attention and pre-trained models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1032–1036, Online. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved Baselines with Visual Instruction Tuning. ArXiv:2310.03744 [cs].

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual Instruction Tuning. ArXiv:2304.08485 [cs].

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

G. Da San Martino, A. Barrón-Cedeño, H. Wachsmuth, R. Petrov, and P. Nakov. 2020. SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. ArXiv:2009.02696 [cs].

Sadiq Muhammed T and Saji K. Mathew. 2022. The disaster of misinformation: a review of research in social media. *International Journal of Data Science and Analytics*, 13(4):271–285.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 Task 3: Detecting the Category, the Framing, and the Persuasion Techniques in Online News in a Multi-lingual Setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Junfeng Tian, Min Gui, Chenliang Li, Ming Yan, and Wenming Xiao. 2021. MinD at SemEval-2021 task 6: Propaganda detection using transfer learning and multimodal fusion. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1082–1087, Online. Association for Computational Linguistics.

SemEval-2024 Task 8: Multidomain, Multimodel and Multilingual Machine-Generated Text Detection

Yuxia Wang,[†] Jonibek Mansurov,[†] Petar Ivanov,[†] Jinyan Su,[†] Artem Shelmanov,[†]
Akim Tsvigun,[†] Osama Mohammed Afzal,[†] Tarek Mahmoud,[†]
Giovanni Puccetti,[§] Thomas Arnold,[¶] Chenxi Whitehouse,[*]
Alham Fikri Aji,[†] Nizar Habash,[†‡] Iryna Gurevych,[†] Preslav Nakov[†]
[†]MBZUAI, UAE [¶]TU Darmstadt, Germany [*]University of Cambridge, UK
[§]Institute of Information Science and Technology, Italy [‡]New York University Abu Dhabi, UAE
{yuxia.wang, jonibek.mansurov, preslav.nakov}@mbzuai.ac.ae

Abstract

We present the results and the main findings of SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Machine-Generated Text Detection. The task featured three subtasks. Subtask A is a binary classification task determining whether a text is written by a human or generated by a machine. This subtask has two tracks: a monolingual track focused solely on English texts and a multilingual track. Subtask B is to detect the exact source of a text, discerning whether it is written by a human or generated by a specific LLM. Subtask C aims to identify the changing point within a text, at which the authorship transitions from human to machine. The task attracted a large number of participants: subtask A monolingual (126), subtask A multilingual (59), subtask B (70), and subtask C (30). In this paper, we present the task, analyze the results, and discuss the system submissions and the methods they used. For all subtasks, the best systems used LLMs.

1 Introduction

The proliferation of Large Language Models (LLMs) has led to a significant increase in the volume of machine-generated text (MGT) across a wide range of domains. This rise has sparked concerns regarding the potential for misuse in fields such as journalism, education, academia, etc (Uchendu et al., 2023; Crothers et al., 2023). Moreover, it poses challenges to maintaining information integrity and ensuring accurate information dissemination. As such, the ability to accurately distinguish between human-written content and machine-generated content has become paramount for identifying potential misuse (Jawahar et al., 2020; Stiff and Johansson, 2022; Macko et al., 2023).

In response to these challenges, we are introducing a shared task that focuses on the detection of machine-generated text across multiple generators, domains, and languages. We are providing large-scale evaluation datasets for **three subtasks** with the primary goals of fostering extensive research in MGT detection, advancing the development of automated systems for detecting MGT, and reducing instances of misuse:

Subtask A: Human vs. Machine Classification. The goal of this subtask is to accurately classify a text as either produced by a human or generated by a machine. This is the basic, but one of the most common use-cases of MGT detection systems for preventing the misuse of LLMs. This task is divided into two tracks: (i) The *monolingual track*, which focuses solely on English texts; and (ii) The *multilingual track*, which involves texts in a variety of languages, thereby expanding the diversity and complexity beyond existing benchmarks.

Subtask B: Multi-Way Generator Detection. This task aims to pinpoint the exact source of a text, i.e., determine whether it originated from a human or a specific LLM (GPT-3, GPT-3.5, GPT-4, Cohere, DALL-E, or BLOOMz). Determining a particular LLM that potentially generated the given text is important from several perspectives: it can help to narrow down the set of LLMs for more sensitive white-box detection techniques or in cases where the generated material is harmful, misleading, or illegal, it might be useful for addressing ethical concerns and legal obligations.

Subtask C: Changing Point Detection. The goal of this subtask is to precisely identify the exact boundary (changing point) within a text at which the authorship transitions from a human to machine happens. The texts begin with human-written content, which at some point is automatically continued by LLMs (GPT and LlaMA series). The percentage of the human-written section varies from 0% to 50%. This task takes into account the fact

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 2057–2079

that in many malignant use-cases of LLMs, the part of the text might be written by a human and a part might be generated by a machine. It is hard to classify a text as machine-generated if a big chunk is actually human-written. This is a way to obscure the usage of LLM, and the formulation of Subtask C addresses this challenge.

The task attracted a large number of participants: 126 teams for the Subtask A monolingual track, 59 teams for the Subtask A multilingual track, 70 teams for Subtask B, and 30 teams for Subtask C, with a total of 54 participating teams having submitted a system description paper for all subtasks.

Next, we introduce the MGT detection techniques considered in this shared task in §2; §3 describes the corpus and the evaluation metrics; §4 details the organization of the task; §5 provides an overview of the participating systems; and §6 discusses the evaluation results.

2 Background

Detecting machine-generated text is primarily formulated as a binary classification task (Zellers et al., 2019; Gehrmann et al., 2019a; Solaiman et al., 2019; Ippolito et al., 2019), naively distinguishing between human-written and machine-generated text. In general, there are two main approaches: the supervised methods (Wang et al., 2024a,b; Uchendu et al., 2021; Zellers et al., 2019; Zhong et al., 2020; Liu et al., 2022) and the unsupervised ones, such as zero-shot methods (Solaiman et al., 2019; Ippolito et al., 2019; Mitchell et al., 2023; Su et al., 2023; Hans et al., 2024). While supervised approaches yield relatively better results, they are susceptible to overfitting (Mitchell et al., 2023; Su et al., 2023). Meanwhile, unsupervised methods may require unrealistic white-box access to the generator. In the following, we provide background information on each subtask, respectively.

Subtask A: Mono-lingual and Multi-lingual Binary Classification Given the prevalence of the binary classification task, various benchmarks assess model performance in both mono-lingual and multi-lingual settings. HC3 (Guo et al., 2023) compares ChatGPT-generated text with human-written text in English and Chinese, utilizing logistic regression models trained on GLTR Test-2 features (Gehrmann et al., 2019a) and RoBERTa (Liu et al., 2019)-based classifiers for detection. Benchmark results by Wang et al. (2024b) include evaluations of several supervised detectors, such as RoBERTa (Liu et al., 2019), XLM-R (Conneau et al., 2019), logistic regression classifier with GLTR features (Gehrmann et al., 2019b), and stylistic features (e.g., stylometry (Li et al., 2014), NELA (Horne et al., 2019) features). Macko et al. (2023) create a similar resource called MULTITuDE for 11 languages in the news domain and conduct an extensive evaluation of various baselines. Our effort extends the previous works by providing evaluation setup for multiple domains, multiple languages, and for state-of-the-art LLMs, including ChatGPT and GPT-4.

Subtask B: Multi-Way Generator Detection Multi-way generator detection, attributing texts not just to their machine-generated nature but also to specific generators, resembles authorship attribution. Munir et al. (2021) find that texts from language models (LMs) have distinguishable features for source attribution. Uchendu et al. (2020) addresses three authorship attribution problems: (1) determining if two texts share the same origin, (2) discerning whether a text is machine or human-generated, and (3) identifying the language model responsible for text generation. Approaches like GPT-who by Venkatraman et al. (2023) employ UID-based features to capture unique signatures of each language model and human author, while Rivera Soto et al. (2024) leverages representations of writing styles.

Subtask C: Change Point Detection Change point detection, which is closely tied to authorship obfuscation (Macko et al., 2024), extends beyond binary/multi-class classification to an adversarial co-authorship setting involving both humans and machines (Dugan et al., 2023). Machine-generated text detection methods are vulnerable to authorship obfuscation attacks such as paraphrasing (Crothers et al., 2022; Krishna et al., 2023; Shi et al., 2023; Koike et al., 2023), back-translation, and change point detection. Related to Subtask C, (Gao et al., 2024) introduces a dataset with mixed machine and human-written texts using operations such as polish, complete (Xie et al., 2023), rewrite (Shu et al., 2023), humanize (adding natural noise (Wang et al., 2021)), and adapt (Gero et al., 2022). Kumarage et al. (2023) uses stylometric signals to quantify changes in tweets and detect when AI starts generating tweets. Different to our task, they focus on human-to-AI author changes within a given Twitter

Split	Source	davinci-003	ChatGPT	Cohere	Dolly-v2	BLOOMz	GPT-4	Machine	Human
Train	Wikipedia	3,000	2,995	2,336	2,702	–	–	11,033	14,497
	Wikihow	3,000	3,000	3,000	3,000	–	–	12,000	15,499
	Reddit	3,000	3,000	3,000	3,000	–	–	12,000	15,500
	arXiv	2,999	3,000	3,000	3,000	–	–	11,999	15,498
	PeerRead	2,344	2,344	2,342	2,344	–	–	9,374	2,357
Dev	Wikipedia	–	–	–	–	500	–	500	500
	Wikihow	–	–	–	–	500	–	500	500
	Reddit	–	–	–	–	500	–	500	500
	arXiv	–	–	–	–	500	–	500	500
	PeerRead	–	–	–	–	500	–	500	500
Test	Outfox	3,000	3,000	3,000	3,000	3,000	3,000	18,000	16,272

Table 1: **Subtasks A: Monolingual Binary Classification.** Data statistics over Train/Dev/Test splits

Split	Language	davinci-003	ChatGPT	LLaMA2	Jais	Other	Machine	Human
Train	English	11,999	11,995	–	–	35,036	59,030	62,994
	Chinese	2,964	2,970	–	–	–	5,934	6,000
	Urdu	–	2,899	–	–	–	2,899	3,000
	Bulgarian	3,000	3,000	–	–	–	6,000	6,000
	Indonesian	–	3,000	–	–	–	3,000	3,000
Dev	Russian	500	500	–	–	–	1,000	1,000
	Arabic	–	500	–	–	–	500	500
	German	–	500	–	–	–	500	500
Test	English	3,000	3,000	–	–	9,000	15,000	13,200
	Arabic	–	1,000	–	100	–	1,100	1,000
	German	–	3,000	–	–	–	3,000	3,000
	Italian	–	–	3,000	–	–	3,000	3,000

Table 2: **Subtasks A: Multilingual Binary Classification.** Data statistics over Train/Dev/Test splits (Others generators are Cohere, Dolly-v2 and BLOOMz)

timeline.

3 Dataset and Metrics

In this section, we describe the datasets and evaluation metrics for all subtask tracks, including the size, domains, generators, and language distribution across training, development, and test splits.

3.1 Subtask A: Monolingual Track

Data: Table 1 presents statistics across generators, domains, and splits. The training set encompasses domains such as Wikipedia, WikiHow, Reddit, arXiv, and PeerRead, comprising a total of 56,400 machine-generated and 63,351 human-written texts. BLOOMz is utilized as an unseen generator in the development set, which contains 2,500 machine-generated and 2,500 human-written texts. For the test set, OUTFOX is introduced as the surprising domain, and GPT-4 serves as the surprising generator, with a dataset of 18,000 machine-generated and 16,272 human-written texts.
Metrics: Accuracy is used to evaluate detectors.

3.2 Subtask A: Multilingual Track

Data: Table 2 presents the dataset statistics. The training set encompasses texts in English, Chinese, Urdu, Bulgarian, and Indonesian, totaling 76,863 machine-generated and 80,994 human-written texts. The development set includes Arabic (sourced from Wikipedia), Russian, and German (sourced from Wikipedia), each contributing 2,000 texts from both machine-generated and human-written sources. In the test set, Italian is introduced as the unexpected language, with *OUTFOX* and *News* serving as new domains for English, Arabic, and German texts. This set comprises 22,100 machine-generated and 20,200 human-written texts.
Metrics: Accuracy is used to evaluate detectors.

Split	Source	davinci-003	ChatGPT	Cohere	Dolly-v2	BLOOMz	Human
Train	Wikipedia	3,000	2,995	2,336	2,702	2,999	3,000
	Wikihow	3,000	3,000	3,000	3,000	3,000	2,995
	Reddit	3,000	3,000	3,000	3,000	2,999	3,000
	arXiv	2,999	3,000	3,000	3,000	3,000	2,998
Dev	PeerRead	500	500	500	500	500	500
Test	Outfox	3,000	3,000	3,000	3,000	3,000	3,000

Table 3: **Subtasks B: Multi-Way Generator Detection.** Data statistics over Train/Dev/Test splits

Domain	Generator	Train	Dev	Test	Total
PeerRead	ChatGPT	**3,649 (232)**	**505 (23)**	**1,522 (89)**	5,676 (344)
	LLaMA-2-7B*	3,649 (5)	505 (0)	**1,035 (1)**	5,189 (6)
	LLaMA-2-7B	3,649 (227)	505 (24)	**1,522 (67)**	5,676 (318)
	LLaMA-2-13B	3,649 (192)	505 (24)	**1,522 (84)**	5,676 (300)
	LLaMA-2-70B	3,649 (240)	505 (21)	**1,522 (88)**	5,676 (349)
OUTFOX	GPT-4	–	–	**1,000 (10)**	1,000 (10)
	LLaMA2-7B	–	–	**1,000 (8)**	1,000 (8)
	LLaMA2-13B	–	–	**1,000 (5)**	1,000 (5)
	LLaMA2-70B	–	–	**1,000 (19)**	1,000 (19)
Total	all	18,245	2,525	11,123	31,893

Table 4: **Subtask C: Change Point Detection.** We use generators GPT and LLaMA-2 series over domains of academic paper review (PeerRead) and student essay (OUTFOX). The number in "()" is the number of examples purely generated by LLMs, i.e., human and machine boundary index=0. LLaMA-2-7B* and LLaMA-2-7B used different prompts. Bold data is used in shared task training, development, and test.

3.3 Subtask B

Data: In Table 3, we incorporate texts from five generators (davinci-003, ChatGPT, Cohere, Dolly-v2, and BLOOMz) alongside human-written texts. The development set features texts from the PeerRead domain, while the test set introduces OUTFOX (specifically, student essays) as the unexpected domain.
Metrics: Accuracy is used to evaluate detectors.

3.4 Subtask C

Data: The training and development sets for subtask C are PeerRead ChatGPT generations, with **5,349** and **505** examples respectively (first row of Table 4), and the test set is the combination of the *test column* of Table 4, totaling 11,123 examples.

Metrics: The Mean Absolute Error (MAE) is used to evaluate the performance of the boundary detection model. It measures the average absolute difference between the predicted position index and the actual changing point.

4 Task Organization

The shared task was run in two phases:

Development Phase. Only training and development data were provided to the participants, with no gold labels available for the development set. Participants competed against each other to achieve the best performance on the development set. A live leaderboard on CodaLab was made available to track all submissions. Teams could make an unlimited number of submissions, and the best score for each team, regardless of the submission time, was displayed in real time on CodaLab.

Test Phase. The test set was released, containing two additional languages—German and Italian for Subtask A Multilingual Track, generator GPT-4 for the Monolingual Track, and a new domain (student essays) for Subtask B. For Subtask C, both new domains and generators were introduced (GPT-4 and LLaMA-2 series based on PeerRead and OUTFOX), which were not disclosed to the participants beforehand (referred to as surprise languages, domains, and generators).

Participants were given approximately three weeks to prepare their predictions. They could submit multiple runs, but they wouldn't receive feedback on their performance. Only the latest submission from each team was considered official and used for the final team ranking.

In total, 125 teams submitted results for Subtask A Monolingual, 62 for Subtask A Multilingual, 70 for Subtask B, and 30 for Subtask C. Additionally, 54 teams submitted system description papers.

After the competition concluded, we released the gold labels for both the development and test sets. Furthermore, we kept the submission system open for the test dataset for post-shared task evaluations and to monitor the state of the art across the different subtasks.

5 Participating Systems

In this section, we first summarize common features for all teams based on the information they provided in the Google Docs. Then, we delve into the methods employed by the top 3 teams, accompanied by brief descriptions of the approaches utilized by the other top 10 teams.

The approaches of all teams are presented in Appendix A.

Team Name	Ranking	small PLM	LLM	GPT	fine-tuning	zero-shot	few-shot (k=?)	Data augmentation	External Data
Genaios	1		✓						
USTC-BUPT	2	✓			✓				
petkaz	12	✓			✓				
HU	17		✓		✓			✓	
TrustAI	20	✓			✓				
L3i++	25		✓		✓				
art-nat-HHU	26	✓			✓				
Unibuc - NLP	28	✓			✓			✓	
NewbieML	30	✓							
QUST	31		✓		✓			✓	
NootNoot	39	✓			✓				
Mast Kalandar	40		✓		✓				
I2C-Huelva	41		✓		✓				
Werkzeug	45	✓			✓				
NCL-UoR	50		✓		✓				
Sharif-MGTD	51		✓		✓	✓			
Collectivized Semantics	62	✓			✓				
SINAI	61		✓		✓				
MasonTigers	71	✓	✓		✓	✓			
DUTh	73		✓		✓				
surbhi	74		✓		✓				
KInIT	77		✓		✓	✓			
RUG-D	100		✓		✓				✓
RUG-5	101	✓	✓		✓				
RUG-3	114		✓	✓	✓				
Mashee	115		✓				2		
RUG-1	117	✓							

Table 5: **Subtask A monolingual** participants methods overview. *small PLM*: Pre-trained Language Model is used, *LLM*: LLM is used, *GPT* indicates if any GPT models are used, *fine-tuning*: applying fine-tuned models, *zero-shot* and *few-shot (k=?)* that zero or more examples are used as demonstrations in in-context learning based on LLMs, *Data augmentation* and *External Data* referes to that augmented data or other external data have been used.

5.1 Monolingual Human vs Machine

Table 5 provides a high-level overview of the methodologies employed by the top-ranking systems in Subtask A monolingual. Most systems utilized either a Pretrained Language Model (PLM) or a Large Language Model (LLM), with the majority of participants fine-tuning their models. Usage of GPT, external data, and few-shot methods was observed in only one team each.

Team Genaios$_{\text{STA_mono:1}}$ (Sarvazyan et al., 2024) achieved the highest performance in this subtask by extracting token-level probabilistic fea-

tures (log probability and entropy) using four LLaMA-2 models: LLaMA-2-7B, LLaMA-2-7B-chat, LLaMA-2-13B, and LLaMA-2-13B-chat. These features were then fed into a Transformer Encoder trained in a supervised manner.

Team USTC-BUPT$_{\text{STA_mono:2}}$ (Guo et al., 2024) secured the second position. Their model is built upon RoBERTa, with the addition of two classification heads: one for binary classification (human or machine) using MLP layers, and another for domain classification (e.g., news, essays, etc.). The latter is equipped with an MLP layer and a gradient reversal layer to enhance transferability between the training and test sets. A sum-up loss is applied, resulting in approximately 8% improvement compared to the RoBERTa baseline.

Team PetKaz$_{\text{STA_mono:12}}$ (Petukhova et al., 2024) utilized a fine-tuned RoBERTa augmented with diverse linguistic features.

In addition to the top three teams: **Team HU$_{\text{STA_mono:17}}$** (Roy Dipta and Shahriar, 2024) employed a contrastive learning-based approach, fine-tuning MPNet on an augmented dataset. **Team TrustAI$_{\text{STA_mono:20}}$** ensembles several classical ML classifiers, Naive Bayes, LightGBM and SGD. **Team L3i++$_{\text{STA\_mono:24}}$** (Tran et al., 2024) investigated various approaches including likelihood, fine-tuning small PLMs, and LLMs, with the latter, fine-tuned LLaMA-2-7B, proving to be the most effective. **Team art-nat-HHU$_{\text{STA_mono:25}}$** (Ciccarelli et al., 2024) utilized a RoBERTa-base model combined with syntactic, lexical, probabilistic, and stylistic features. **Team Unibuc - NLP$_{\text{STA_mono:28}}$** (Marchitan et al., 2024) jointly trained Subtasks A and B based on RoBERTa. Most other teams fine-tuned either RoBERTa or XLM-RoBERTa for MGT detection, enhancing the models through various techniques, ranging from a mixture of experts by **Team Werkzeug$_{\text{STA_mono:45}}$** (Wu et al., 2024) to low-rank adaptation by **Team NCL-UoR$_{\text{STA_mono:50}}$** (Xiong et al., 2024), while **Team Sharif-MGTD$_{\text{STA_mono:51}}$** (Ebrahimi et al., 2024) preferred careful fine-tuning of PLMs alone.

5.2 Multilingual Human vs Machine

Table 6 provides an overview of the methods employed by the top-performing systems for Subtask A Multilingual. Various techniques are utilized, including zero-shot learning based on LLMs, PLM-based classifiers, and ensemble models.

Team USTC-BUPT$_{\text{STA_Multi:1}}$ (Guo et al., 2024) secured the top position. They initially detect the language of the input text. For English text, they average embeddings from LLaMA-2-70B, followed by classification through a two-stage CNN. For texts in other languages, the classification problem is transformed into fine-tuning a next-token prediction task using the mT5 model, incorporating special tokens for classification. Their approach integrates both monolingual and multilingual strategies, exploiting large language models for direct embedding extraction and model fine-tuning. This enables the system to adeptly handle text classification across a diverse range of languages, especially those with fewer resources.

Team Name	Ranking	small PLM	LLM	GPT	Fine-tuning	Zero-shot	Data augmentation	External Data
USTC-BUPT	1		✓		✓			
FI Group	2	✓			✓			
KInIT	3		✓		✓	✓		
L3i++	5		✓		✓			
QUST	6		✓		✓		✓	
AIpom	9		✓		✓			
SINAI	21		✓		✓			
Unibuc-NLP	22	✓			✓			
Werkzeug	30	✓			✓			
RUG-5	32	✓	✓		✓			
DUTh	33		✓		✓			
RUG-D	39		✓		✓			✓
MasonTigers	49	✓	✓		✓	✓		
TrustAI	55	✓			✓			

Table 6: **Subtask A multilingual** participants methods.

Team FI Group$_{\text{STA_Multi:2}}$ (Ben-Fares et al., 2024) implemented a hierarchical fusion strategy that adaptively combines representations from different layers of XLM-RoBERTa-large, moving beyond the conventional "[CLS]" token classification to sequence labeling for enhanced detection of stylistic nuances.

Team KInIT$_{\text{STA_Multi:3}}$ (Spiegel and Macko, 2024) combined fine-tuned LLMs with zero-shot statistical methods, employing a two-step majority voting system for predictions. Their method emphasizes language identification, per-language threshold calibration, and the integration of both fine-tuned and statistical detection methods, demonstrating the power of ensemble strategies. For the LLMs, they utilized QLoRA PEFT to fine-tune

Team Name	Ranking	small PLM	LLM	GPT	fine-tuning	zero-shot	Data augmentation
AISPACE	1		✓		✓		✓
Unibuc - NLP	2	✓			✓		✓
USTC-BUPT	3		✓				
L3i++	6		✓		✓		
MLab	7				✓		
Werkzeug	8	✓			✓		
TrustAI	14	✓			✓		
MGTD4ADL	17				✓		✓
scalar	18	✓					✓
UMUT	23	✓			✓		
QUST	36		✓		✓		✓
MasonTigers	38	✓	✓		✓	✓	
RUG-5	41	✓	✓		✓		
RUG-D	44		✓		✓		
DUTh	49		✓		✓		
clulab-UofA	62		✓	✓	✓		✓

Table 7: **Subtask B** Participants method overview.

Falcon-7B and Mistral-7B.

Other teams explored various approaches, like using LoRA-finetuned LLMs as classifiers (**Team AIpom$_{\text{STA_Multi:9}}$**) (Shirnin et al., 2024), using semantic and syntactic aspects of the texts (**RFBES$_{\text{STA_Multi:10}}$**) (Heydari Rad et al., 2024) or fusing perplexity with text and adding a classification head (**Team SINAI$_{\text{STA_Multi:21}}$**) (Gutiérrez Megías et al., 2024). Each team's method provides insights into the complexities of multilingual text detection, ranging from the use of specific LLMs and PLMs to the use of linguistic and probabilistic metrics and ensemble techniques (Wu et al., 2024; Brekhof et al., 2024; Kyriakou et al., 2024; Puspo et al., 2024; Urlana et al., 2024).

5.3 Multi-way Detection

Table 7 provides an overview of the approaches employed by the top-ranking systems for Subtask B. Similar to Subtask A, most solutions do not use GPT and zero-shot approaches. The best-performing solutions primarily exploit LLMs and data augmentation.

Team AISPACE$_{\text{STB:1}}$ (Gu and Meng, 2024) achieved the highest performance in this subtask by fine-tuning various encoder and encoder-decoder models, including RoBERTa, DeBERTa, XLNet, Longformer, and T5. They augmented the data with instances from Subtask A and explored the effects of different loss functions and learning rate values. Their method leverages a weighted Cross-Entropy loss to balance samples in different classes and uses an ensemble of fine-tuned models to improve robustness.

Team Unibuc - NLP$_{\text{STB:2}}$ (Marchitan et al., 2024) employed a Transformer-based model with a unique two-layer feed-forward network as a classification head. They also augmented the data with instances from the Subtask A monolingual dataset.

Team USTC-BUPT$_{\text{STB:3}}$ (Guo et al., 2024) leveraged LLaMA-2-70B to obtain token embeddings and applied a three-stage classification. They first distinguished human-generated from machine-generated text using LLaMA-2-70B, then categorized ChatGPT and Cohere as one class and distinguished them from davinci-003, BLOOMz, and Dolly-v2. Finally, they performed binary classification between ChatGPT and Cohere.

Team L3i++$_{\text{STB:6}}$ (Tran et al., 2024) conducted a comparative study among three groups of methods: metric-based models, fine-tuned classification language models (RoBERTa, XLM-R), and a fine-tuned LLM, LLaMA-2-7B, finding LLaMA-2 to outperform other methods. They analyzed errors and various factors in their paper.

Team MLab$_{\text{STB:7}}$ (Li et al., 2024) fine-tuned DeBERTa and analyzed the embeddings from the last layer to provide insights into the embedding space of the model.

Team Werkzeug$_{\text{STB:8}}$ (Wu et al., 2024) utilized RoBERTa-large and XLM-RoBERTa-large to encode text, addressing the problem of anisotropy in text embeddings produced by pre-trained language models (PLMs) by introducing a learnable parametric whitening (PW) transformation. They also used multiple PW transformation layers as experts under the mixture-of-experts (MoE) architecture to capture features of LLM-generated text from different perspectives.

Other teams explored various approaches, including different loss functions and sentence transformers (**Team MGTD4ADL$_{\text{STB:17}}$**) (Chen et al., 2024), RoBERTa fine-tuning (**Team UMUTeam$_{\text{STB:23}}$**) (pan et al., 2024), stacking ensemble techniques (**Team MasonTigers$_{\text{STB:38}}$**) (Puspo et al., 2024), and basic ML models with linguistic-stylistic features (**Team RUG-5$_{\text{STB:41}}$**)

Team Name	Ranking	small PLM	LLM	LSTM (+) CNN	fine-tuning	Data augmentation	CRF layer
TM-TREK$_{\text{STC:1}}$	1	✓			✓		✓
AIpom$_{\text{STC:2}}$	2		✓		✓		
USTC-BUPT$_{\text{STC:3}}$	3	✓	✓		✓	✓	
RKadiyala$_{\text{STC:6}}$	6	✓			✓		✓
DeepPavlov$_{\text{STC:7}}$	7	✓			✓	✓	
RUG-5 $_{\text{STC:17}}$	17	✓	✓		✓		
TueCICL$_{\text{STC:22}}$	22			✓			
jelarson$_{\text{STC:25}}$	25						
MasonTigers $_{\text{STC:27}}$	27	✓					
Unibuc-NLP$_{\text{STC:28}}$	28			✓	✓		

Table 8: **Subtask C** Participants method overview.

(Darwinkel et al., 2024).

5.4 Boundary Identification

Table 8 presents an overview of the methods used by the top-ranking systems for Subtask C. The best performing solutions are mainly based on ensemble strategies, with some employing data augmentation.

Team TM-TREK$_{\text{STC:1}}$ (Qu and Meng, 2024) achieved the highest performance in Subtask C. They utilized an ensemble of XLNet models, each trained with a distinct seed, and used a straightforward voting mechanism on the output logits. They also explored the integration of LSTM and CRF layers on top of various PLMs, along with continued pretraining and fine-tuning of PLMs, and dice loss functions to enhance model performance.

Team AIpom$_{\text{STC:2}}$ (Shirnin et al., 2024) introduced a novel two-stage pipeline merging outputs from an instruction-tuned, decoder-only (Mistral-7B-OpenOrca) model and two encoder-only sequence taggers.

Team USTC-BUPT$_{\text{STC:3}}$ (Guo et al., 2024) fine-tuned a DeBERTa model with data augmentation and framed the task as a token classification problem.

Team RKadiyala $_{\text{STC:6}}$ (Kadiyala, 2024) fine-tuned various encoder-based models with a Conditional Random Field (CRF) layer and found Deberta-V3 to perform the best on the development set.

Team DeepPavlov $_{\text{STC:7}}$ (Voznyuk and Konovalov, 2024) fine-tuned the Deberta-v3 model using the provided dataset and developed a data preprocessing pipeline for data augmentation.

Other teams explored diverse CNN, LSTM (**Team TueCICL $_{\text{STC:22}}$**) (Stuhlinger and Winkler, 2024), (**Team Unibuc - NLP $_{\text{STC:28}}$**) (Marchitan et al., 2024), and regression-based (**Team jelarson $_{\text{STC:25}}$**) (Larson and Tyers, 2024) techniques to address this challenge, although many did not surpass the baselines due to issues related to model overfitting or inadequate word embeddings.

6 Results and Discussion

6.1 Subtask A

There were three submissions for subtask A, which were submitted in time, but had the wrong file name, which prevented us from scoring them automatically. We eventually manually fixed the names and scored them, and we also added them to the ranking but marked them with a *. They should be considered as unofficial submissions.

Monolingual Table 9 presents the performance of systems in the monolingual track of Subtask A. Out of 125 participating teams, 15 surpassed the baseline, with the top-performing team (Genaios) achieving an accuracy of 96.88. Notably, several teams demonstrated high precision and recall scores, indicating robust performance in distinguishing between human-generated and machine-generated text in a binary classification context.

Multilingual Table 10 presents the performance of systems in the multilingual track of Subtask A, where Team USTC-BUPT emerges as the top performer among 62 participating teams, achieving an accuracy of 95.99, remarkably close to the English-only result. Their methodology entails a blend of language detection and fine-tuning tasks using LLaMA-2-70B for English and the mT5 model for others, showcasing their adaptability across diverse languages.

Similarly, among the 22 teams surpassing the baseline, the majority leverage advanced LLMs such as LLaMA, Mistral, etc., while also emphasizing syntax and writing style differences between human and AI-generated texts. For example, Team FI Group implements a hierarchical fusion strategy to adaptively fuse representations from different BERT layers, prioritizing syntax over semantics for improved classification accuracy. Likewise, Team KInIT employs an ensemble approach, combining fine-tuned LLMs with zero-shot statistical methods,

Rank	Team	Prec	Recall	F1-score	Acc	Rank	Team	Prec	Recall	F1-score	Acc
*	dianchi	96.21	99.19	97.68	97.53	62	Collectivized Semantics	68.21	99.39	80.90	75.35
1	Genaios	96.11	98.03	97.06	96.88	63	IUCL	68.13	98.33	80.49	74.96
2	USTC-BUPT	95.75	96.86	96.30	96.10	64	annedadaa	68.01	97.69	80.19	74.66
3	mail6djj	94.87	97.18	96.02	95.76	65	cmy99	67.92	97.96	80.22	74.62
4	howudoin	93.48	98.12	95.74	95.42	66	xiaoll	67.92	97.96	80.22	74.62
5	idontknow	94.57	95.42	94.99	94.72	67	SINAI	67.31	99.88	80.42	74.46
6	seven	90.12	98.31	94.04	93.46	68	yuwert777	68.78	92.96	79.06	74.14
7	zongxiong	93.54	93.82	93.68	93.35	69	yaoxy	68.78	92.96	79.06	74.14
8	mahsaamani	90.59	96.23	93.32	92.77	70	moniszcz	67.25	97.66	79.65	73.79
9	bennben	91.49	95.05	93.24	92.76	71	MasonTigers	67.59	95.72	79.23	73.64
10	infinity2357	91.92	90.96	91.43	91.05	72	AT	67.30	96.59	79.33	73.56
11	AISPACE	84.76	99.92	91.72	90.52	73	DUTh	66.27	99.92	79.69	73.24
12	petkaz	85.54	98.59	91.61	90.51	74	surbhi	69.38	87.40	77.35	73.12
13	moniszcz1	86.96	95.68	91.11	90.20	75	thanet	69.47	86.69	77.13	73.00
14	moniszcz3	86.96	95.68	91.11	90.20	76	Kathlalu	74.47	73.89	74.18	72.98
15	flash	82.39	99.77	90.25	88.68	77	KInIT	66.14	98.44	79.12	72.71
*	baseline	93.36	84.02	88.44	88.47	78	iimasNLP	67.81	87.08	76.25	71.50
16	ericmxf	81.71	99.98	89.93	88.24	79	wwzzhh	64.38	99.49	78.18	70.82
17	HU	82.63	97.24	89.34	87.81	80	bharathsk	64.48	98.69	78.00	70.76
18	jrutkowski2	84.58	93.44	88.79	87.61	81	apillay2	64.48	98.69	78.00	70.76
19	lihaoran	89.26	85.86	87.53	87.15	82	ashinee20	71.91	71.57	71.74	70.39
20	TrustAI	89.21	85.50	87.31	86.95	83	longfarmer	63.81	99.03	77.61	69.99
21	TM-TREK	79.47	99.99	88.56	86.43	84	mlnick	63.61	100	77.76	69.96
22	jojoc	86.30	87.76	87.02	86.25	85	vasko	63.47	99.93	77.63	69.75
23	RFBES	91.58	80.64	85.76	85.95	86	Groningen F	72.74	65.62	68.99	69.02
24	L3i++	81.41	94.66	87.53	85.84	87	hhy123	62.76	99.94	77.11	68.83
25	art-nat-HHU	86.29	86.04	86.17	85.49	88	1024m	62.54	99.98	76.95	68.53
26	FI Group	79.52	96.99	87.39	85.30	89	lhy123	62.54	99.96	76.94	68.53
27	phuhoang	87.72	83.65	85.64	85.26	90	thang	62.25	99.98	76.73	68.14
28	Unibuc-NLP	78.01	99.86	87.59	85.14	91	nikich28	61.90	99.26	76.25	67.52
29	sushvin	82.65	89.76	86.06	84.73	92	niceone	61.70	98.39	75.84	67.08
30	NewbieML	79.32	95.06	86.48	84.39	93	pmalesa	60.73	97.87	74.95	65.64
31	QUST	76.88	99.91	86.89	84.17	94	mahaalblooki	60.85	96.23	74.56	65.51
32	MLab	83.17	85.83	84.48	83.44	95	bertsquad	60.32	98.94	74.95	65.27
33	ziweizheng	82.31	85.01	83.63	82.53	96	jjonczyk	60.25	99.31	75.00	65.23
34	AIpom	74.34	99.97	85.27	81.86	97	dkoterwa	60.12	99.98	75.09	65.16
35	lyaleo	79.38	87.82	83.39	81.62	98	lystsoval	92.30	35.31	51.07	64.48
36	yunhfang	75.26	95.31	84.10	81.08	99	sunilgundapu	59.22	99.31	74.20	63.72
37	sankalpbahad	78.22	88.08	82.86	80.86	100	RUG-D	59.19	99.35	74.18	63.68
38	aktsvigun	78.22	88.08	82.86	80.86	101	RUG-5	60.79	84.13	70.58	63.17
39	NootNoot	78.22	88.08	82.86	80.86	102	harshul24	54.55	57.14	55.81	62.00
40	Mast Kalandar	74.65	96.16	84.05	80.83	103	basavraj10	54.55	57.14	55.81	62.00
41	I2C-Huelva	73.92	98.01	84.28	80.79	104	samnlptaskab	58.01	99.93	73.41	61.97
42	priority497	73.31	99.69	84.49	80.78	105	partnlu	57.87	99.96	73.31	61.76
43	wjm123	73.31	99.69	84.49	80.78	106	teams2024	57.78	99.97	73.23	61.61
44	scalar	73.10	99.97	84.45	80.67	107	Rkadiyala	57.30	99.98	72.85	60.86
45	werkzeug	75.28	93.88	83.56	80.59	108	rtuora	57.18	99.89	72.73	60.66
46	blain	72.51	99.96	84.05	80.07	109	teamlanlp2	56.93	99.71	72.48	60.24
47	xxm981215	72.32	99.79	83.86	79.83	110	jakubbebacz	56.89	99.88	72.49	60.18
48	moyanxinxu	72.32	99.79	83.86	79.83	111	dandread	56.19	99.87	71.92	59.04
49	jrutkowskikag1	73.02	97.54	83.52	79.78	112	pask1	55.97	99.80	71.72	58.67
50	NCL-UoR	75.10	90.84	82.22	79.37	113	skillissue	55.14	100	71.09	57.27
51	Sharif-MGTD	73.41	93.75	82.35	78.89	114	RUG-3	54.92	99.73	70.83	56.87
52	wgm123	71.16	99.94	83.13	78.69	115	Mashee	57.11	59.58	58.32	55.27
53	logiczmaksimka	70.79	99.29	82.65	78.11	116	TueCICL	55.37	69.61	61.68	54.57
54	somerandomjj	70.43	98.55	82.15	77.51	117	RUG-1	52.52	100	68.87	52.52
55	totylkokuba	70.43	98.55	82.15	77.51	118	novice8	52.24	68.68	59.34	50.57
56	lly123	69.25	99.95	81.81	76.66	119	ronghaopan	52.49	38.47	44.40	48.70
57	mimkag2	69.69	97.99	81.45	76.56	120	kamer	52.89	29.56	37.93	48.48
58	priyansk	69.28	99.36	81.64	76.53	121	helenpy	75.29	1.79	3.51	48.11
59	nampfiev1995	77.92	76.93	77.43	76.44	122	ascisel	7.14	0.01	0.01	47.44
60	xiangrunli	68.23	99.97	81.11	75.54	123	laida	40.18	19.28	26.06	42.53
61	roywang	68.23	99.97	81.11	75.54	124	nz28555	40.31	33.18	36.40	39.10
*	badrock	71.13	89.50	79.27	75.41						

Table 9: **Subtask A monolingual** Prec (precision), Recall, and F1-scores(%) with respect to **MGT**.

Rank	Team	Prec	Recall	F1-score	Acc
1	USTC-BUPT	94.93	97.53	96.21	95.99
2	FI Group	94.28	98.00	96.10	95.85
3	KInIT	92.95	97.86	95.34	95.00
4	priyansk	90.70	98.14	94.28	93.77
5	L3i++	92.47	94.00	93.23	92.87
6	QUST	90.45	90.98	90.71	90.27
7	xxm981215	90.45	90.98	90.71	90.27
8	NCL-UoR	81.42	95.41	87.86	86.23
9	AIpom	80.72	95.80	87.61	85.85
10	RFBES	85.43	85.27	85.35	84.71
11	blain	76.12	98.67	85.94	83.14
12	xiangrunli	75.20	99.67	85.73	82.66
13	wgm123	75.20	99.67	85.73	82.66
14	roywang	75.08	99.75	85.68	82.58
15	logiczmaksimka	74.34	99.33	85.04	81.74
16	zaratiana	74.75	96.68	84.31	81.21
17	thanet	76.18	92.56	83.58	81.00
*	baseline	73.45	99.30	84.44	80.89
18	cmy99	73.29	99.61	84.45	80.83
19	lly123	73.09	99.67	84.33	80.65
20	moyanxinxu	73.09	99.67	84.33	80.65
21	SINAI	72.51	99.91	84.04	80.17
22	Unibuc-NLP	71.82	99.79	83.52	79.43
23	annedadaa	72.16	98.57	83.32	79.39
24	1024m	71.03	99.91	83.03	78.66
25	sunilgundapu	71.04	98.86	82.67	78.35
26	hirak	70.79	99.66	82.78	78.34
27	bertsquad	70.45	99.12	82.36	77.82
28	Rkadiyala	69.99	99.94	82.33	77.59
*	dianchi	69.88	99.91	82.24	77.46
29	lyaleo	69.50	99.83	81.95	77.03
30	werkzeug	69.33	99.81	81.82	76.83
31	mlnick	69.25	99.81	81.77	76.75
32	RUG-5	69.90	96.78	81.17	76.55
33	DUTh	68.95	99.93	81.60	76.45
34	dandread	68.31	99.75	81.09	75.69
35	Genaios	68.30	99.73	81.07	75.67
36	vasko	67.99	98.68	80.50	75.03
37	thang	67.16	99.78	80.29	74.40
38	mahsaamani	68.53	93.21	78.98	74.09
39	RUG-D	65.03	99.55	78.67	71.79
40	omarnasr	64.80	99.21	78.40	71.43
41	lhy123	64.62	99.85	78.46	71.36
42	priority497	64.62	99.85	78.46	71.36
43	hhy123	64.47	99.84	78.35	71.17
44	wjm123	64.47	99.84	78.35	71.17
45	aktsvigun	62.83	99.36	76.98	68.96
46	sankalpbahad	62.83	99.36	76.98	68.96
47	NootNoot	62.83	99.36	76.98	68.96
48	nampfiev1995	61.37	77.15	68.36	62.70
49	MasonTigers	56.77	100	72.42	60.21
50	RUG-1	51.33	97.39	67.23	51.22
51	novice8	51.95	84.56	64.36	51.08
52	scalar	52.04	80.17	63.11	51.04
53	mahaalblooki	48.96	51.24	50.08	50.55
54	Sharif-MGTD	51.42	67.13	58.23	50.53
55	TrustAI	51.15	62.04	56.07	50.06
56	sky2024just	51.96	26.37	34.99	48.79
57	laida	48.88	20.56	28.94	47.27

Table 10: **Subtask A multilingual** Prec (precision), Recall, and F1-scores(%) with respect to **MGT**.

Rank	Team	Prec	Recall	F1-score	Acc
1	AISPACE	91.81	90.85	90.84	90.85
2	Unibuc - NLP	88.69	86.96	87.03	86.96
3	USTC-BUPT	89.54	84.33	82.72	84.33
4	dianchi	86.45	83.48	83.62	83.48
5	NootNoot	86.68	83.12	83.15	83.12
6	L3i++	86.01	83.12	83.08	83.12
7	MLab	85.00	82.67	82.76	82.67
8	werkzeug	86.30	82.23	81.63	82.23
9	flash	88.29	82.23	79.46	82.23
10	juse7198	86.83	82.03	80.72	82.03
11	idontknow	88.44	80.94	77.47	80.94
12	TM-TREK	86.42	79.84	79.46	79.84
13	howudoin	80.02	79.68	79.79	79.68
14	TrustAI	83.80	79.19	79.07	79.19
15	I2C-Huelva	84.45	78.90	78.82	78.90
16	ericmxf	85.52	78.74	76.88	78.74
17	MGTD4ADL	83.78	76.96	74.46	76.96
18	scalar	81.90	76.26	76.00	76.26
19	ronghaopan	81.11	75.19	71.38	75.35
20	sunilgundapu	81.06	75.06	73.81	75.06
*	baseline	81.14	74.61	72.59	74.61
21	Collectivized Semantics	82.35	73.87	70.26	73.87
22	priyansk	78.06	73.36	67.05	73.36
23	logiczmaksimka	67.73	69.13	64.36	69.13
24	annedadaa	79.55	68.98	64.55	68.98
25	hhy123	65.94	67.77	63.01	67.77
26	xiangrunli	65.94	67.77	63.01	67.77
27	wjm123	65.94	67.77	63.01	67.77
28	lhy123	65.94	67.77	63.01	67.77
29	lly123	65.94	67.77	63.01	67.77
30	wgm123	65.94	67.77	63.01	67.77
31	moyanxinxu	65.94	67.77	63.01	67.77
32	priority497	65.94	67.77	63.01	67.77
33	thang	66.36	67.68	63.79	67.68
34	blain	63.15	67.23	62.35	67.23
35	xxm981215	65.77	67.21	62.41	67.21
36	QUST	65.77	67.21	62.41	67.21
37	mahaalblooki	63.72	66.27	61.82	66.27
38	MasonTigers	73.62	65.04	64.47	65.04
39	Rkadiyala	65.81	64.91	59.98	64.91
40	1024m	66.10	64.38	59.82	64.38
41	RUG-5	62.21	64.21	59.04	64.21
42	thanet	63.42	61.88	55.58	61.88
43	mlnick	66.84	61.79	57.53	61.79
44	RUG-D	66.39	61.54	53.82	61.54
45	Groningen F	60.10	60.84	57.90	60.84
46	NCL-UoR	69.03	60.15	58.05	60.15
47	mahsaamani	60.41	59.42	52.89	59.42
48	dandread	71.95	58.35	52.28	58.35
49	DUTh	63.71	56.68	51.25	56.68
50	bertsquad	57.27	55.97	51.49	55.97
51	RUG-3	61.51	54.23	49.26	54.23
52	cmy99	58.04	53.35	50.58	53.35
53	skysky12	60.86	53.31	50.14	53.31
54	vasko	59.98	52.82	50.38	52.82
55	phuhoang	61.54	50.79	50.21	50.79
56	rtuora	54.21	50.32	44.15	50.32
57	AT	53.13	48.59	43.91	48.59
58	teams2024	45.50	47.01	41.05	47.01
59	windwind22	39.87	39.31	32.79	39.31
60	helenpy	39.88	38.27	32.20	38.27
61	iimasNLP	39.88	38.27	32.20	38.27
62	clulab-UofA	37.53	29.29	24.58	29.29
63	samnlptaskab	25.78	27.81	21.07	27.81
64	mhr2004	17.06	17.06	17.06	17.06
65	xiaoll	5.73	17.15	8.47	17.02
66	surbhi	17.24	16.77	15.10	16.77
67	roywang	2.78	16.67	4.76	16.67
68	RUG-1	2.78	16.67	4.76	16.67
69	novice8	16.39	16.55	13.93	16.55
70	NewbieML	15.99	15.58	14.13	15.30

Table 11: **Subtask B: Multi-Way Generator Detection** Prec (precision), Recall, and F1-scores(%) macro average.

resulting in a unique combination of techniques that effectively enhances classification accuracy.

Overall, these successful methodologies under-

Rank	Team	MAE	Rank	Team	MAE
1	TM-TREK	15.68	16	mahaalblooki	25.95
2	AIpom	15.94	17	RUG-5	26.07
3	USTC-BUPT	17.70	18	mahsaamani	26.27
4	ywnh111	18.08	19	aktsvigun	26.40
5	ywnh222	18.51	20	skillissue	27.99
6	Rkadiyala	18.54	21	NootNoot	28.01
7	DeepPavlov	19.25	22	TueCICL	34.88
8	knk42	19.42	23	dandread	35.17
9	vasko	19.93	24	novice8	44.82
10	logiczmaksimka	19.93	25	jelarson	48.14
11	AISPACE	21.19	26	TueSents	58.95
*	baseline	21.54	27	MasonTigers	60.78
12	ericmxf	21.55	28	Unibuc - NLP	74.28
13	blain	21.80	29	lanileqiu	78.18
14	1024m	22.36	30	scalar	87.72
15	cmy99	24.68			

Table 12: **Subtask C: Boundary Identification**.

score the importance of leveraging advanced LLMs, ensemble techniques, and comprehensive analysis to achieve superior performance in detecting machine-generated text across multilingual contexts.

6.2 Subtask B

For Subtask B (Multi-Way detection), 70 teams participated, with 20 surpassing the baseline of 74.61 accuracy. Table 11 displays the full results. In summary, the subtask results underline the effectiveness of diverse and innovative approaches, including fine-tuning advanced models (e.g., RoBERTa, DeBERTa, XLNet, Longformer, T5), data augmentation (e.g., using Subtask A instances), ensemble strategies, and the exploration of novel loss functions and learning techniques. The leading entries showcased a range of methodologies, from leveraging the power of large language models and addressing embedding anisotropy to integrating traditional and neural methods, underscoring the dynamic and evolving nature of NLP research. For instance, Team AISPACE utilized a weighted Cross-Entropy loss and an ensemble approach based on model performance per class, which led to the highest accuracy of 90.85.

6.3 Subtask C

Of the 30 systems that were submitted for Subtask C, 11 outperformed the baseline MAE of 21.54. The top system, TM-TREK, achieved the best submitted MAE of 15.68. A significant majority of the top-performing teams relied on ensembles of PLMs, indicating a consensus that combining the strengths of multiple models can lead to more robust and accurate predictions. This approach leverages the diverse representations and strengths of different models to mitigate weaknesses inherent in individual systems.

Data augmentation emerged as a critical strategy among leading teams, suggesting its effectiveness in enhancing model performance by providing a richer, more varied training dataset. This includes both the generation of new training examples and the manipulation of existing data to better capture the complexity and variability of natural language.

Despite the advanced methodologies deployed, some teams struggled with issues related to overfitting and the adequacy of word embeddings. This underscores the ongoing challenges in developing models that generalize well to unseen data and the critical role of embeddings in capturing semantic and syntactic nuances of language.

7 Conclusion and Future Work

We have described SemEval-2024 Task 8 on Multigenerator, Multidomain, and Multilingual Machine-Generated Text Detection. The task garnered significant interest from researchers, with 126, 59, 70, and 30 teams submitting entries for Subtask A Monolingual, Subtask A Multilingual, Subtask B, and Subtask C, respectively. Additionally, we received 54 system description papers before finalizing this submission.

Overall, Subtasks A and B were relatively easier, with all systems showing improvements over the baseline. However, Subtask C proved to be significantly more challenging. Fewer teams participated, and many struggled to surpass our baseline results set in (Wang et al., 2024a).

In future work, we plan to extend our focus beyond machine-generated text detection to other modalities such as image, speech, and video detection. Additionally, we intend to develop an open-source demonstration system capable of distinguishing between AI-generated content and human-produced content.

Limitations

Despite providing a comprehensive dataset that spans multiple languages, generators, and domains across three distinct tasks in machine-generated text detection, our study encounters several limitations that pave the way for future research.

Firstly, the reliance on textual data without access to white-box information, such as token-level probabilities, confines our detection methods to

black-box approaches across all tasks. These methods might exhibit reduced effectiveness and struggle to generalize across new domains, generators, and languages. Additionally, they are susceptible to language-style attacks, including paraphrasing in different tones, back-translation, and other forms of textual adversarial tactics. In contrast, methods that leverage watermarking and white-box patterns show greater promise for robust MGT detection.

Secondly, our approach to boundary identification presupposes that each text comprises an initial segment written by humans followed by machine-generated content, with only one transition point. However, real-world scenarios often present more complex challenges. It is crucial not only to ascertain the presence of mixed text but also to identify all transition points. Texts may originate from human authors and undergo refinement via machine assistance, or vice versa, encompassing machine generation followed by human revision. Addressing these nuanced scenarios will be a focus of our future research efforts.

Ethics and Broader Impact

This section outlines potential ethical considerations related to our work.

Data Collection and Licenses Our study utilizes pre-existing corpora, specifically the M4 and OUTFOX datasets, which have been publicly released for research purposes under clear licensing agreements.

Security Implications The dataset underpinning our shared task aims to foster the development of robust MGT detection systems. These systems are crucial for identifying and mitigating misuse scenarios, such as curbing the proliferation of automated misinformation campaigns and protecting individuals and institutions from potential financial losses. In fields such as journalism, academia, and legal proceedings, where the authenticity of information is of utmost importance, MGT detection plays a vital role in maintaining content integrity and trust. Furthermore, by enhancing public awareness of the capabilities and limitations of LLMs, we can cultivate a healthy skepticism towards digital content. Effective MGT detection mechanisms are essential for ensuring that users can place their trust in content generated by LLMs.

Acknowledgments

We extend our deepest gratitude to the SemEval Shared Task 2024 organizing committee for their enduring patience and support throughout our task's development, and to all participants for their innovative contributions and collaborative spirit during the task coordination phase. Our thanks also go to the anonymous reviewers and program committee chairs, whose constructive feedback has significantly contributed to the improvement of our paper.

References

Pranjal Aggarwal and Deepanshu Sachdeva. 2024. Cunlp at semeval-2024 task 8: Classify human and ai generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1–6, Mexico City, Mexico. Association for Computational Linguistics.

Huseyin Alecakir, Puja Chakraborty, Pontus Henningsson, Matthijs van Hofslot, and Alon Scheuer. 2024. Groningen team a at semeval-2024 task 8: Human/machine authorship attribution using a combination of probabilistic and linguistic features. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1931–1937, Mexico City, Mexico. Association for Computational Linguistics.

Jainit Bafna, Hardik Mittal, Suyash Sethia, Manish Shrivastava, and Radhika Mamidi. 2024. Mast kalandar at semeval-2024 task 8: On the trail of textual origins: Roberta-bilstm approach to detect ai-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1638–1644, Mexico City, Mexico. Association for Computational Linguistics.

Sankalp Bahad, Yash Bhaskar, and Parameswari Krishnamurthy. 2024. Fine-tuning language models for ai vs human generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 905–908, Mexico City, Mexico. Association for Computational Linguistics.

Maha Ben-Fares, Urchade Zaratiana, Simon Hernandez, and Pierre Holat. 2024. Fi group at semeval-2024 task 8: A syntactically motivated architecture for multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1155–1160, Mexico City, Mexico. Association for Computational Linguistics.

Thijs Brekhof, Xuanyi Liu, Yuwen Zhou, and Joris Ruitenbeek. 2024. Groningen team d at semeval-2024 task 8: Exploring data generation and a combined model for fine-tuning llms for multidomain machine-generated text detection. In *Proceedings of*

the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 378–385, Mexico City, Mexico. Association for Computational Linguistics.

Lujia Cao, Ece Lara Kilic, and Katharina Will. 2024. Kathlalu at semeval-2024 task 8: A comparative analysis of binary classification methods for distinguishing between human and machine-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 386–389, Mexico City, Mexico. Association for Computational Linguistics.

Huixin Chen, Jan Büssing, David Rügamer, and Ercong Nie. 2024. Team mgtd4adl at semeval-2024 task 8: Leveraging (sentence) transformer models with contrastive learning for identifying machine-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1722–1729, Mexico City, Mexico. Association for Computational Linguistics.

Vittorio Ciccarelli, Cornelia Genz, Nele Mastracchio, Wiebke Petersen, Anna Stein, and Hanxin Xia. 2024. Team art-nat-hhu at semeval-2024 task 8: Stylistically informed fusion model for mgt-detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1701–1708, Mexico City, Mexico. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Evan Crothers, Nathalie Japkowicz, Herna Viktor, and Paula Branco. 2022. Adversarial robustness of neural-statistical features in detection of generative transformers. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*.

Patrick Darwinkel, Sijbren van Vaals, Marieke van der Holt, and Jarno van Houten. 2024. Groningen group e at semeval-2024 task 8: Detecting machine-generated texts through pre-trained language models augmented with explicit linguistic-stylistic features. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 995–1003, Mexico City, Mexico. Association for Computational Linguistics.

Ayan Datta, Aryan Chandramania, and Radhika Mamidi. 2024. Weighted layer averaging roberta for black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1634–1637, Mexico City, Mexico. Association for Computational Linguistics.

Rina Donker, Björn Overbeek, Dennis van Thulden, and Oscar Zwagers. 2024. Groningen team f at semeval-2024 task 8: Detecting machine-generated text using feature-based machine learning models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1924–1930, Mexico City, Mexico. Association for Computational Linguistics.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12763–12771.

Seyedeh Fatemeh Ebrahimi, Karim Akhavan Azari, Amirmasoud Iravani, Arian Qazvini, Pouya Sadeghi, Zeinab Taghavi, and Hossein Sameti. 2024. Sharif-mgtd at semeval-2024 task 8: A transformer-based approach to detect machine generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 552–559, Mexico City, Mexico. Association for Computational Linguistics.

Chujie Gao, Dongping Chen, Qihui Zhang, Yue Huang, Yao Wan, and Lichao Sun. 2024. Llm-as-a-coauthor: The challenges of detecting llm-human mixcase. *arXiv preprint arXiv:2401.05952*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019a. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019b. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for science writing using language models. In *Designing interactive systems conference*, pages 1002–1019.

Renhua Gu and Xiangfeng Meng. 2024. Aispace at semeval-2024 task 8: A class-balanced soft-voting system for detecting multi-generator machine-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1487–1492, Mexico City, Mexico. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *CoRR*, abs/2301.07597.

Zikang Guo, Kaijie Jiao, Xingyu Yao, Yuning Wan, Haoran Li, Benfeng Xu, Licheng Zhang, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2024. Ustcbupt at semeval-2024 task 8: Enhancing machine-generated text detection via domain adversarial neural networks and llm embeddings. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1522–1533, Mexico City, Mexico. Association for Computational Linguistics.

Alberto Gutiérrez Megías, L. Alfonso Ureña-López, and Eugenio Martínez Cámara. 2024. Sinai at semeval-2024 task 8: Fine-tuning on words and perplexity as features for detecting machine written text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1516–1521, Mexico City, Mexico. Association for Computational Linguistics.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*.

Mohammad Heydari Rad, Farhan Farsi, Shayan Bali, Romina Etezadi, and Mehrnoush Shamsfard. 2024. Rfbes at semeval-2024 task 8: Investigating syntactic and semantic features for distinguishing ai-generated and human-written texts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 437–441, Mexico City, Mexico. Association for Computational Linguistics.

Benjamin D Horne, Jeppe Nørregaard, and Sibel Adali. 2019. Robust fake news detection over time and attack. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1):1–23.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Ganesh Jawahar, Muhammad Abdul Mageed, and VS Laks Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309.

Ram Mohan Rao Kadiyala. 2024. Rkadiyala at semeval-2024 task 8: Black-box word-level text boundary detection in partially machine generated texts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 498–506, Mexico City, Mexico. Association for Computational Linguistics.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2023. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. *arXiv preprint arXiv:2307.11729*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *arXiv preprint arXiv:2303.13408*.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023. Stylometric detection of ai-generated text in twitter timelines. *arXiv preprint arXiv:2303.03697*.

Theodora Kyriakou, Ioannis Maslaris, and Avi Arampatzis. 2024. Duth at semeval 2024 task 8: Comparing classic machine learning algorithms and llm based methods for multigenerator, multidomain and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1069–1075, Mexico City, Mexico. Association for Computational Linguistics.

Joseph Larson and Francis Tyers. 2024. Team jelarson at semeval 2024 task 8: Predicting boundary line between human and machine generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 464–471, Mexico City, Mexico. Association for Computational Linguistics.

Jenny S Li, John V Monaco, Li-Chiou Chen, and Charles C Tappert. 2014. Authorship authentication using short messages from social networking sites. In *2014 IEEE 11th International Conference on e-Business Engineering*, pages 314–319. IEEE.

Kevin Li, Kenan Hasanaliyev, Sally Zhu, George Altshuler, Alden Eberts, Eric Chen, Kate Wang, Emily Xia, Eli Browne, Ian Chen, and Umut Eren. 2024. Team mlab at semeval-2024 task 8: Analyzing encoder embeddings for detecting llm-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1474–1478, Mexico City, Mexico. Association for Computational Linguistics.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2022. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *arXiv preprint arXiv:2212.10341*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Anand Kumar M, Abhin B, and Sidhaarth Murali. 2024. Scalar at semeval-2024 task 8: Unmasking the machine : Exploring the power of roberta ensemble for detecting machine generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1124–1128, Mexico City, Mexico. Association for Computational Linguistics.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Dominik Macko, Robert Moro, Adaku Uchendu, Ivan Srba, Jason Samuel Lucas, Michiharu Yamashita, Nafis Irtiza Tripto, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2024. Authorship obfuscation in multilingual machine-generated text detection. *arXiv preprint arXiv:2401.07867*.

Teodor-George Marchitan, Claudiu Creanga, and Liviu P. Dinu. 2024. Team unibuc - nlp at semeval-2024 task 8: Transformer and hybrid deep learning based models for machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 390–398, Mexico City, Mexico. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *CoRR*, abs/2301.11305.

Shaoor Munir, Brishna Batool, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar. 2021. Through the looking glass: Learning to attribute synthetic text generated by language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1811–1822.

ronghao pan, José Antonio García-Díaz, Pedro José Vivancos-Vicente, and Rafael Valencia-García. 2024. Umuteam at semeval-2024 task 8: Combining transformers and syntax features for machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 683–688, Mexico City, Mexico. Association for Computational Linguistics.

Kseniia Petukhova, Roman Kazakov, and Ekaterina Kochmar. 2024. Petkaz at semeval-2024 task 8: Can linguistics capture the specifics of llm-generated text? In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1129–1136, Mexico City, Mexico. Association for Computational Linguistics.

Valentin Pickard and Hoa Do. 2024. Tuesents at semeval-2024 task 8: Predicting the shift from human authorship to machine-generated output in a mixed text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 816–819, Mexico City, Mexico. Association for Computational Linguistics.

Srikar Kashyap Pulipaka, Shrirang Mhalgi, Joseph Larson, and Sandra Kübler. 2024. Semeval task 8: A comparison of traditional and neural models for detecting machine authored text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1015–1020, Mexico City, Mexico. Association for Computational Linguistics.

Sadiya Sayara Chowdhury Puspo, Md Nishat Raihan, Dhiman Goswami, Al Nahian Bin Emran, Amrita Ganguly, and Özlem Uzuner. 2024. Masontigers at semeval-2024 task 8: Performance analysis of transformer-based models on machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1354–1362, Mexico City, Mexico. Association for Computational Linguistics.

Xiaoyan Qu and Xiangfeng Meng. 2024. Tm-trek at semeval-2024 task 8: Towards llm-based automatic boundary detection for human-machine mixed text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 696–701, Mexico City, Mexico. Association for Computational Linguistics.

Areeg Fahad Rasheed and M. Zarkoosh. 2024. Mashee at semeval-2024 task 8: The impact of samples quality on the performance of in-context learning for machine text classification. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 60–63, Mexico City, Mexico. Association for Computational Linguistics.

MohammadHossein Rezaei, Yeaeun Kwon, Reza Sanayei, Abhyuday Singh, and Steven Bethard. 2024. Clulab-uofa at semeval-2024 task 8: Detecting machine-generated text using triplet-loss-trained text similarity and text classification. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1509–1515, Mexico City, Mexico. Association for Computational Linguistics.

Rafael Rivera Soto, Kailin Koch, Aleem Khan, Barry Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. *arXiv e-prints*, pages arXiv–2401.

alberto rodero, Jacinto Mata, and Victoria Pachón Álvarez. 2024. Boosting ai-generated text detection with multimodal models and optimized ensembles. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 832–839, Mexico City, Mexico. Association for Computational Linguistics.

Shubhashis Roy Dipta and Sadat Shahriar. 2024. Hu at semeval-2024 task 8a: Can contrastive learning learn embeddings to detect machine-generated text? In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 472–478, Mexico City, Mexico. Association for Computational Linguistics.

Areg Mikael Sarvazyan, José Ángel González, and Marc Franco-Salvador. 2024. Genaios at semeval-2024 task 8: Detecting machine-generated text by mixing language model probabilistic features. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 101–107, Mexico City, Mexico. Association for Computational Linguistics.

Surbhi Sharma and Irfan Mansuri. 2024. Team innovative at semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1161–1165, Mexico City, Mexico. Association for Computational Linguistics.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2023. Red teaming language model detectors with language models. *arXiv preprint arXiv:2305.19713*.

Alexander Shirnin, Nikita Andreev, Vladislav Mikhailov, and Ekaterina Artemova. 2024. Aipom at semeval-2024 task 8: Detecting ai-produced outputs in m4. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1678–1683, Mexico City, Mexico. Association for Computational Linguistics.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Canoee Liu, Simon Tong, Jindong Chen, and Lei Meng. 2023. Rewritelm: An instruction-tuned large language model for text rewriting. *arXiv preprint arXiv:2305.15685*.

Marco Siino. 2024. Badrock at semeval-2024 task 8: Distilbert to detect multigenerator, multidomain and multilingual black-box machine-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 239–245, Mexico City, Mexico. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Michal Spiegel and Dominik Macko. 2024. Kinit at semeval-2024 task 8: Fine-tuned llms for multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 545–551, Mexico City, Mexico. Association for Computational Linguistics.

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 13(4):363–383.

Daniel Stuhlinger and Aron Winkler. 2024. Tuecicl at semeval-2024 task 8: Resource-efficient approaches for machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1608–1612, Mexico City, Mexico. Association for Computational Linguistics.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Bao Tran and Nhi Tran. 2024. Newbieml at semeval-2024 task 8: Ensemble approach for multidomain machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 347–353, Mexico City, Mexico. Association for Computational Linguistics.

Hanh Thi Hong Tran, Tien Nam Nguyen, Antoine Doucet, and Senja Pollak. 2024. L3i++ at semeval-2024 task 8: Can fine-tuned large language model detect multigenerator, multidomain, and multilingual black-box machine-generated text? In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 13–21, Mexico City, Mexico. Association for Computational Linguistics.

Adaku Uchendu, Thai Le, and Dongwon Lee. 2023. Attribution and obfuscation of neural text authorship: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 25(1):1–18.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8384–8395.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296*.

Ashok Urlana, Aditya Saibewar, Bala Mallikarjunarao Garlapati, Charaka Vinayak Kumar, Ajeet Singh, and Srinivasa Rao Chalamala. 2024. Trustai at semeval-2024 task 8: A comprehensive analysis of multidomain machine generated text detection techniques. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 914–921, Mexico City, Mexico. Association for Computational Linguistics.

Andric Valdez, Fernando Márquez, Jorge Pantaleón, Helena Gómez, and Gemma Bel-Enguix. 2024. iimasnlp at semeval-2024 task 8: Unveiling structure-aware language models for automatic generated text identification. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1099–1103, Mexico City, Mexico. Association for Computational Linguistics.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector. *arXiv preprint arXiv:2310.06202*.

Anastasia Voznyuk and Vasily Konovalov. 2024. Deeppavlov at semeval-2024 task 8: Leveraging transfer learning for detecting boundaries of machine-generated texts. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1833–1841, Mexico City, Mexico. Association for Computational Linguistics.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. 2024a. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. *arXiv preprint arXiv:2402.11175*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Yuchen Wei. 2024. Team at at semeval-2024 task 8: Machine-generated text detection with semantic embeddings. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 479–483, Mexico City, Mexico. Association for Computational Linguistics.

Youlin Wu, Kaichun Wang, Kai Ma, Liang Yang, and Hongfei LIN. 2024. Werkzeug at semeval-2024 task 8: Llm-generated text detection via gated mixture-of-experts fine-tuning. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 534–539, Mexico City, Mexico. Association for Computational Linguistics.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. 2023. The next chapter: A study of large language models in storytelling. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 323–351.

Feng Xiong, Thanet Markchom, Ziwei Zheng, Subin Jung, Varun Ojha, and Huizhi Liang. 2024. Ncl-uor at semeval-2024 task 8: Fine-tuning large language models for multigenerator, multidomain, and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 163–169, Mexico City, Mexico. Association for Computational Linguistics.

Xiaoman Xu, Xiangrun Li, Taihang Wang, Jianxiang Tian, and Ye Jiang. 2024. Team qust at semeval-2024 task 8: A comprehensive study of monolingual and multilingual approaches for detecting ai-generated text. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 450–457, Mexico City, Mexico. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9051–9062.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. *arXiv preprint arXiv:2010.07475*.

Appendix

A Method Summary

A.1 Monolingual Human vs, Machine

Team Genaios$_{\text{STA_mono:1}}$ (Sarvazyan et al., 2024) achieves the highest accuracy on Subtask A – Monolingual by extracting token-level probabilistic features using four Llama-2 models: Llama-2-7b, Llama-2-7b-chat, Llama-2-13b, and Llama-2-13b-chat. For each token they compute the log probability of the observed token, the log probability of the token predicted by each of the language models, and the entropy of the distribution. These features are then fed to a Transformer Encoder trained in a supervised fashion to detect synthetic text.

Team USTC-BUPT$_{\text{STA_mono:2}}$ (Guo et al., 2024) incorporates domain adversarial neural networks into the task of machine-generated text detection to reach the second position in the ranking of Subtask A – Monolingual. They add a gradient reversal layer on top of the baseline, a supervised classifier based on RoBERTa. In addition, they exploit domain labels to enhance the transferability of learning between training and testing datasets. Their architecture is based on RoBERTa and adds two classification heads, one for category classification (human or synthetic) and one for domain classification (e.g. news, essays, etc.), the former uses an MLP layer and the latter is composed of an MLP together with a gradient reversal layer. Finally, the loss is also adapted by summing together the category and the domain losses. The submission evaluated an improvement of approximately 8% compared to the baseline.

Team PetKaz$_{\text{STA_mono:12}}$ (Petukhova et al., 2024) uses a PLM, RoBERTa-base, fine-tuned for synthetic text detection and enhances it with linguistic features, to train a feed-forward binary classifier (human or synthetic). Their final model uses diverse features and notably, they undersample the human data.

Team HU$_{\text{STA_mono:17}}$ (Roy Dipta and Shahriar, 2024) Adopts an architecture trained with a contrastive learning approach based on fine-tuning *sentence-transformers/all-mpnet-base-v2*. The model is trained on an augmented dataset obtained by paraphrasing sentences in the training set.

Team TrustAI$_{\text{STA_mono:20}}$ (Urlana et al., 2024) tries two approaches: (a) an ensemble approach with the combination of Multinomial Naive Bayes, LGBM Classifier (lightGBM classifier) and SGD classifier. Each is trained on the concatenation of tf-idf and spaCy embeddings obtained from the Subtask A – Monolingual dataset and (b) a synthetic text classifier based on RoBERTa fine-tuned first with the outputs of the 1.5B-parameter GPT-2 model and subsequently on the Subtask A – monolingual dataset. They show that exploring methodologies with different assumptions helps identify the best performing approach.

Team RFBES$_{\text{STA_mono:23}}$ (Heydari Rad et al., 2024) Both semantic and syntactic considerations were taken into account. For semantic analysis, emphasis was placed on smaller text segments rather than the entire document, operating under the belief that AI models could produce similarly coherent long texts as humans. To achieve this, the XLM-RoBERTa model was employed. Regarding syntactic analysis, a stacked bidirectional LSTM model was used to categorize texts based on their grammatical patterns using UPOS tags. Interestingly, no significant differences in UPOS tag distribution between AI-generated and human-written texts were revealed by the findings.

Team L3i++$_{\text{STA\_mono:24}}$ (Tran et al., 2024) Proposes a comparative study among 3 groups of methods to detect synthetic texts: 5 likelihood-based methods; 2 fine-tuned sequence-labeling language models (RoBERTa, XLM-RoBERTa); and a fine-tuned large language model, llama-2-7b. LLaMA 2 outperforms the rest and accurately detects machine-generated texts.

Team art-nat-HHU$_{\text{STA_mono:25}}$ (Ciccarelli et al., 2024) fine-tunes a RoBERTa model pre-trained for AI-detection and combines it with a set of linguistic features: syntactic, lexical, probabilistic and stylistic. To improve the classifier, they train two separate neural networks on these features, one for each class predicted by the RoBERTa-based classifier.

Team Unibuc - NLP$_{\text{STA_mono:28}}$ (Marchitan et al., 2024) fine-tunes a Transformer-based model with a MLP as a classification head. They combine the datasets of Subtask A – monolingual and Subtask B to obtain a larger training set.

NewbieML$_{\text{STA_mono:30}}$ (Tran and Tran, 2024) embeds texts with Longformer-large. Then they Ensemble SVM, LogisticRegression and XGBoost with, as a meta model, a KNN.

Team QUST$_{\text{STA_mono:31}}$ (Xu et al., 2024) experiments with multiple models on a dataset extended through data augmentation. They select the two best-performing models for ensembling: (1) a fine-tuned RoBERTa model, combined with the Multiscale Positive-Unlabeled (MPU) training and (2) a DeBERTa model. They use these two for model fusion through stacking ensemble.

Team NootNoot$_{\text{STA_mono:39}}$ (Bahad et al., 2024) carefully fine-tunes a RoBERTa-base model to classify human written and synthetic texts.

Team Mast Kalandar$_{\text{STA_mono:40}}$ (Bafna et al., 2024) trains a classifier that uses a frozen RoBERTa model with an LSTM head to classify human vs machine written texts.

Team I2C-Huelva$_{\text{STA_mono:41}}$ (rodero et al., 2024) proposes a method to use multimodal models together with text analysis to enhance synthetic text detection. To mix the two approaches they explore ensemble by testing several voting methods.

Team Werkzeug$_{\text{STA_mono:45}}$ (Wu et al., 2024) uses Roberta-large and XLM-roberta-large to encode texts. To address the anitostropic embedding space created by transformer-based language models, they employ several learnable parametric whitening (PW) transformation. They show that addressing the anisotropicity of the embedding space improves accuracy in detecting synthetic text.

Team NCL-UoR$_{\text{STA_mono:50}}$ (Xiong et al., 2024) fine-tunes several PLMs including XLM-RoBERTa, RoBERTa with Low-Rank Adaptation (LoRA) and DistilmBERT. Finally, they use majority voting ensembling with XLM-RoBERTa and LoRA-RoBERTa. To confirm that ensembling is a strong technique to boost synthetic text classification accuracy.

Team Sharif-MGTD$_{\text{STA_mono:51}}$ (Ebrahimi et al., 2024) carefully fine-tunes RoBERTa-base for synthetic text detection, show that pre-trained language models are a versatile approach.

Team BadRock$_{\text{STA_mono:*}}$ (Siino, 2024) is based on a fine-tuning of a DistilBERT trained on the SST-2 dataset.

Team Collectivized Semantics$_{\text{STA_mono:62}}$ (Datta et al., 2024) fine-tunes Roberta-base using Ada-LoRa and uses the weighted sum of all the layer hidden states' mean as features to train a classifier. They show that exploiting the knowledge at all layers of encoder language models helps when detecting synthetic texts.

Team IUCL$_{\text{STA_mono:63}}$ (Pulipaka et al., 2024) tries both classical ML classifiers, Naive Bayes and Decision Trees as well as fine-tuning transformers and they conclude that fine-tuned RoBERTa is best among the methods they try.

Team SINAI$_{\text{STA_mono:67}}$ (Gutiérrez Megías et al., 2024) compares three methods: (a) supervised classification, based on fine-tuning the XLM-RoBERTa-Large language model; (b) likelihood-based methods, using GPT-2 to compute the perplexity of each text and use this perplexity as a score; (c) a hybrid approach that merges text with its perplexity value into a classification head. The choice of a mixed approach proves effective in improving synthetic text detection accuracy.

Team MasonTigers$_{\text{STA_mono:71}}$ (Puspo et al., 2024) experiments with different transformer-based models: Roberta, DistilBERT, ELECTRA and ensembles these models. They also experiment with zero-shot prompting and finetuning FlanT5. Further confirming that ensembling is a strong methodology for detecting synthetic texts.

Team AT$_{\text{STA_mono:72}}$ (Wei, 2024) adopts three different semantic embedding algorithms, GLOVE, n-gram embeddings and SentenceBERT as well as their concatenation. The author finds that these pre-trained embeddings, while fast to compute, are not as effective as a fine-tuned RoBERTa model.

Team DUTh$_{\text{STA_mono:73}}$ (Kyriakou et al., 2024) experiments with several supervised classification models based on PLMs. Finally, they opt for a fine-tuned mBERT trained for 5 epochs. This approach shows how PLMs fine-tuning is a versatile approach that can be effective when detecting synthetic texts.

Team surbhi$_{\text{STA_mono:74}}$ (Sharma and Mansuri, 2024) creates two sets of features (a) stylometric features based on the length of text, the number of words, the average length of words, the number of short words, the proportion of digits and capital letters, individual letters and digits frequencies, hapax-legomena, a measure of text richness, and the frequency of 12 punctuation marks and (b) n-grams: frequencies of the 100 most frequent character-level bi-grams and tri-grams; (c) the output probabilities of fine-tuned Roberta model. Each set of features is used to train a classifier and finally, stylometric and n-gram features are chosen as the best-performing ones. They prove that more classical features can still be valuable when attempting the detection of synthetic text.

Team Kathlalu$_{\text{STA_mono:76}}$ (Cao et al., 2024) investigates two methods for constructing a binary classifier to distinguish between human-generated and machine-generated text. The main emphasis is on a straightforward approach based on Zipf's law, which, despite its simplicity, achieves a moderate level of performance. Additionally, they briefly discuss experimentation with the utilization of unigram word counts.

Team KInIT$_{\text{STA_mono:77}}$ (Spiegel and Macko, 2024) uses two approaches: (a) an ensemble using two-step majority voting for predictions, consisting of 2 LLMs (Falcon-7B and Mistral-7B) fine-tuned using the train set only; (b) 3 zero-shot statistical methods (Entropy, Rank, Binoculars) using Falcon-7B and Falcon-7B-Instruct for calculating the metrics. For classification they use per-language threshold calibration, showing that likelihood-based methods are a viable solution to detect machine-written texts.

Team iimasNLP$_{\text{STA_mono:78}}$ (Valdez et al., 2024) fine-tune 4 different language models to identify human and machine generated text, ERNIE, SpanBERT, ConvBERT and XLNet. They find out that RoBERTa is a stronger classifier. In general this shows how fine-tuning PLMs is an effective approach to identify synthetic text.

Team Groningen-F$_{\text{STA_mono:86}}$ (Donker et al., 2024) leverage features including tense of the sentence, the voice of the sentence, the sentiment of the sentence, and the number of pronouns vs. proper nouns on the basis of SVM and FFNN models. The hypothesis here is that traditional models may generalize better than LLMs. It is more computationally effective than LLMs.

Team RUG-D$_{\text{STA_mono:100}}$ (Brekhof et al., 2024) fine-tunes different DeBERTa models on a dataset extended with additional synthetic samples. Showing that PLMs fine-tuning is a versatile approach that can be effective in the detection of synthetic texts.

Team RUG-5$_{\text{STA_mono:101}}$ (Darwinkel et al., 2024) fine-tunes different pre-trained models for synthetic text classification, distilbert-base-cased for the monolingual tasks and distilbert-base-multilingual-cased for the multilingual ones. Moreover, they explore the use of a Random Forest classifier using frozen distilbert-base-cased embeddings concatenated with 20 linguistic and stylistic features. This approach shows how choosing the right PLMs is crucial for better performance in a given task.

Team Mashee$_{\text{STA_mono:115}}$ (Rasheed and Zarkoosh, 2024) selects high-quality and low-quality samples using a Chi-square test and adopts the selected samples for few-shot classification using the FlanT5-Large language model. This approach shows how few-shot methodologies can benefit from a careful example selection.

Team TueCICL$_{\text{STA_mono:116}}$ (Stuhlinger and Winkler, 2024) uses a Charachter-level LSTM with pre-trained word2vec embeddings as input to train synthetic text detector. Doing so, they show how one does not necessarily have to use transformers.

Team RUG-1$_{\text{STA_mono:117}}$ (Alecakir et al., 2024) combines a linear model with document-level features and token-level features that are first passed through an LSTM. Through this methodology, they leverage both local (token-level) and global (document-level) information to identify human-written and synthetic texts.

Team CUNLP$_{\text{STA_mono:unknown}}$ [1] (Aggarwal and Sachdeva, 2024) involved employing a range of machine learning techniques, including logistic regression, transformer models, attention mechanisms, and unsupervised learning methods. Through rigorous experimentation, they identified key features influencing classification accuracy, namely text length, vocabulary richness, and coherence. Notably, the highest classification accuracy was achieved by integrating transformer models with TF-IDF representation and feature engineering. However, it is essential to note that this approach demanded substantial computational resources due to the complexity of transformer models and the incorporation of TF-IDF. Additionally, their investigation encompassed a thorough exploration of various ML algorithms, extensive hyperparameter tuning, and optimization techniques. Furthermore, they conducted detailed exploratory data analysis to gain insights into the structural and lexical characteristics of the text data.

A.2 Multilingual Human vs Machine

Team USTC-BUPT$_{\text{STA_Multi:1}}$ (Guo et al., 2024) secured the top position. They initially detect the language of the input text. For English text, they average embeddings from Llama-2-70B, followed by classification through a two-stage CNN. For texts in other languages, the classification problem is transformed into fine-tuning a next-token prediction task using the mT5 model, incorporating special tokens for classification. Their approach integrates both monolingual and multilingual strategies, exploiting large language models for direct embedding extraction and model fine-tuning. This enables the system to adeptly handle text classification across a diverse range of languages, especially those with fewer resources.

Team FI Group$_{\text{STA_Multi:2}}$ (Ben-Fares et al., 2024) came in second place. Their methodology began with analyzing latent space distinctions between human and AI-generated texts using Sentence-BERT, hypothesizing that syntax and writing style differences are key. They utilized a hierarchical fusion strategy to adaptively fuse representations from different BERT layers, focusing on syntax over semantics. By classifying each token as Human or AI, their model captures detailed text structures, leveraging the XLM-RoBERTa-Large model for robust multilingual performance.

Team KInIT$_{\text{STA_Multi:3}}$ (Spiegel and Macko, 2024) placed third by employing an ensemble of two fine-tuned LLMs (Falcon-7B and Mistral-7B) and three zero-shot statistical methods, using a two-step majority voting system. This unique combination of fine-tuned and statistical methods, complemented by language identification and per-language threshold calibration, showcases their innovative approach to integrating diverse techniques for enhanced classification accuracy.

Team L3i++$_{\text{STA\_Multi:5}}$ (Tran et al., 2024) explored a comparative study among metric-based models, fine-tuned sequence-labeling language models, and a large-scale LLM, finding LLaMA-2 to outperform others in detecting machine-generated texts. Their methodological diversity and comprehensive analysis underline the strengths of fine-tuning LLMs for complex classification tasks across languages.

Team QUST$_{\text{STA_Multi:6}}$ (Xu et al., 2024) employed a fine-tuned XLM-RoBERTa model within a stacking ensemble framework, incorporating the MPU framework and DeBERTa model. Their approach emphasizes the efficacy of model fusion and fine-tuning on a multilingual dataset, highlighting the potential of ensemble strategies in enhancing model performance.

Team AIpom$_{\text{STA_Multi:9}}$ (Shirnin et al., 2024) utilized a LoRA-Finetuned LLM for classifying texts as real or fake, achieving notable results with a limited dataset. Their unique approach of using an LLM as a classifier, despite an accidental label swap during training, emphasizes the versatility and potential of LLMs in unconventional scenarios.

Team RFBES$_{\text{STA_Multi:10}}$ (Heydari Rad et al., 2024) Both semantic and syntactic considerations were taken into account. For semantic analysis, emphasis was placed on smaller text segments rather than the

[1] Team CUNLP submitted results for development set, but no submissions for the test set, resulting unknown valid rank.

entire document, operating under the belief that AI models could produce similarly coherent long texts as humans. To achieve this, the XLM-RoBERTa model was employed. Regarding syntactic analysis, a stacked bidirectional LSTM model was used to categorize texts based on their grammatical patterns using UPOS tags. Interestingly, no significant differences in UPOS tag distribution between AI-generated and human-written texts were revealed by the findings.

Team SINAI$_{\text{STA_Multi:21}}$ (Gutiérrez Megías et al., 2024) compared various systems before settling on a fusion model that integrates text with perplexity values for classification. Their comprehensive approach, blending fine-tuning with innovative use of perplexity, offers insightful perspectives on leveraging multiple data dimensions for classification.

Team Unibuc-NLP$_{\text{STA_Multi:22}}$ (Marchitan et al., 2024) focused on exploring different methods of layer selection and fine-tuning within a transformer-based architecture. Their pursuit of optimizing layer interactions for classification tasks highlights the importance of fine-tuning strategies in achieving model effectiveness.

Team Werkzeug$_{\text{STA_Multi:30}}$ (Wu et al., 2024) applied parametric whitening transformations under a mixture-of-experts architecture to address text embedding anisotropy issues. Their methodological innovation, aimed at capturing a broader range of language styles, underscores the potential of advanced architectures in improving classification accuracy.

Team RUG-5$_{\text{STA_Multi:32}}$ (Darwinkel et al., 2024) augmented DistilBERT with an additional layer for classification, exploring linguistic-stylistic features alongside Random Forest classifiers. Their approach of blending traditional ML techniques with PLMs offers a novel perspective on enhancing text classification through feature integration.

Team DUTh$_{\text{STA_Multi:33}}$ (Kyriakou et al., 2024) compared machine learning algorithms and LLMs, ultimately selecting a fine-tuned XLM-RoBERTa model. Their comparative analysis provides valuable insights into the effectiveness of different methodologies for text classification tasks.

Team RUG-D$_{\text{STA_Multi:39}}$ (Brekhof et al., 2024) used an ensemble of monolingual and multilingual models, testing the performance impact of additional training data. Their ensemble approach and data augmentation strategy highlight the importance of model and data selection in optimizing classification performance.

Team MasonTigers$_{\text{STA_Multi:49}}$ (Puspo et al., 2024) experimented with different transformer models and finetuning strategies, showcasing the effectiveness of ensembling and fine-tuning in addressing classification challenges.

Team TrustAI$_{\text{STA_Multi:55}}$ (Urlana et al., 2024) focused on fine-tuning the bert-base-multilingual-cased model, demonstrating the potential of pre-trained models in multilingual text classification tasks.

A.3 Multi-way Detection

Team AISPACE$_{\text{STB:1}}$ (Gu and Meng, 2024) achieves the highest performance in this subtask by fine-tuning various encoder and encoder-decoder models, including RoBERTa, DeBERTa, XLNet, Longformer, and T5. They augment the data with instances from Subtask A and explore the effects of different loss functions and learning rate values. Based on this analysis, they leverage a weighted Cross-Entropy loss to balance samples in different classes. Furthermore, they use an ensemble of different fine-tuned models to improve the robustness of the system. The weights of the models in the ensemble are assigned based on their performance on each class rather than their performance on the whole accuracy.

Team Unibuc - NLP$_{\text{STB:2}}$ (Marchitan et al., 2024) use a Transformer-based model with a peculiar two-layer feed-forward network as a classification head. They also augment the data with instances from Subtask A monolingual dataset.

Team USTC-BUPT$_{\text{STB:3}}$ (Guo et al., 2024) first leverage the 'Llama-2-70B' model to obtain embeddings of the tokens in the text and then average them across all tokens. Next, they employ a three-stage classification approach using the CNN classifier.

Firstly, they distinguish between human-generated and machine-generated text using the Llama-2-70B model. Secondly, they categorize ChatGPT and Cohere as a single class for a four-class classification, differentiating them from Davinci, Bloomz, and Dolly. Finally, they perform a binary classification

between ChatGPT and Cohere. Despite solid performance, their method does not require fine-tuning.

Team L3i++$_{\text{STB:6}}$ (Tran et al., 2024) conduct a comparative study among three groups of methods: metric-based models, fine-tuned classification language models (RoBERTa, XLM-R), and a fine-tuned LLM, LLaMA-2-7b. They find LLaMA-2 outperforming the methods from the other groups in MGT detection. The team reveals the analysis of errors and various factors in their paper.

Team MLab$_{\text{STB:7}}$ (Li et al., 2024) fine-tune DeBERTa and analyze the embeddings from the last layer. They provide insights into the embedding space of the model.

Team Werkzeug$_{\text{STB:8}}$ (Wu et al., 2024) utilizes RoBERTa-large and XLM-RoBERTa-large to encode the text. They tackle the problem of anisotropy in text embeddings produced by pre-trained language models (PLMs) by introducing a learnable parametric whitening (PW) transformation. Furthermore, to capture the features of LLM-generated text from different perspectives, they use multiple PW transformation layers as experts under the mixture-of-experts (MoE) architecture equipped with a gating router in their final solution.

Team TrustAI$_{\text{STB:14}}$ (Urlana et al., 2024) explore different pretrained and statistical models for detecting synthetic text, ultimately selecting the RoBERTa-base OpenAI Detector for its effectiveness. This model, originally fine-tuned with outputs from the 1.5B-parameter GPT-2 model, is further fine-tuned on the Subtask-B dataset.

Team MGTD4ADL$_{\text{STB:17}}$ (Chen et al., 2024) combine traditional Transformer models (RoBERTa-base, RoBERTa-large, GPT-2-small, XLNet, T5-small) with Sentence Transformers(all-mpnet-base-v2 and all-roberta-large-v1). They further diversify their approach by leveraging different data augmentation techniques and experimenting with various loss functions such as Cross-Entropy (CE), Supervised Contrastive Learning (SCL), and Dual Contrastive Loss (DUALCL).

Team scalar$_{\text{STB:18}}$ (M et al., 2024) employ an ensemble of three RoBERTa-base models using an individual validation set for each model.

Team UMUTeam$_{23}$ (pan et al., 2024) use fine-tuned RoBERTa model combined with syntactic features of the text such as word length, part of speech, function word frequency, stop-word ratio, and sentence length.

Team QUST$_{\text{STB:36}}$ (Xu et al., 2024) use fine-tuned RoBERTa and DeBERTa models, integrating them through a stacking ensemble technique.

Team MasonTigers$_{\text{STB:38}}$ (Puspo et al., 2024) implement an ensemble of 3 PLMs: RoBERTa, DeBERTa, and ELECTRA. Additionally, they employ zero-shot prompting and use a fine-tuned FLAN-T5 model.

Team RUG-5$_{\text{STB:41}}$ (Darwinkel et al., 2024) expands the architecture of DistilBERT models by adding an additional classification layer that incorporates 20 linguistic-stylistic features. They also explore the use of Random Forest classifier on top of embeddings from DistilBERT combined with the same set of linguistic-stylistic features.

Team RUG-D$_{\text{STB:44}}$ (Brekhof et al., 2024) focus on fine-tuning DeBERTa models.

Team Groningen-F$_{\text{STB:45}}$ (Donker et al., 2024) trained traditional machine learning models (SVM and FFNN) with features including tense of the sentence, the voice of the sentence, the sentiment of the sentence, and the number of pronouns vs. proper nouns.

Team DUTh$_{\text{STB:49}}$ (Kyriakou et al., 2024) explore traditional machine learning algorithms along with BERT for their task. Ultimately, they proceed with BERT fine-tuned for 5 epochs.

Team AT$_{\text{STB:58}}$ (Wei, 2024) adopts three different semantic embedding algorithms, GLOVE, n-gram embeddings and SentenceBERT as well as their concatenation to identify the generator in Subtask B. The author finds that these pre-trained embeddings, while fast to compute, are not as effective as a fine-tuned RoBERTa model.

Team iimasNLP$_{\text{STB:61}}$ (Valdez et al., 2024) fine-tune 4 different language models to classify text generated by different models: ERNIE, SpanBERT, ConvBERT and XLNet. They find out that RoBERTa is a stronger classifier. In general this shows how fine-tuning PLMs is an effective approach to identify the generator model.

Team CLULab-UofA$_{\text{STB:62}}$ (Rezaei et al., 2024) combine LLM fine-tuning with contrastive learning, specifically using triplet loss.

A.4 Boundary Identification

Team TM-TREK$_{\text{STC:1}}$ (Qu and Meng, 2024) achieved the highest performance in Subtask C by employing an ensemble of models including Longformer, Bigbird, and XLNet for long-text sequence labeling. A simple voting mechanism was used to aggregate the output logits. Their innovative strategy also involved integrating LSTM and CRF layers atop various pre-trained language models (PLMs), along with continuous pretraining, fine-tuning, and utilizing dice loss functions to enhance model performance.

Team AIpom$_{\text{STC:2}}$ (Shirnin et al., 2024) introduced a two-stage pipeline that combines outputs from an instruction-tuned, decoder-only model (Mistral-7B-OpenOrca) with two encoder-only sequence taggers. Initially, they trained an instruction-tuned autoregressive model to insert a [BREAK] token into input texts, delineating human-written parts from machine-generated ones. Subsequently, these annotated texts were processed by an encoder-based model for sequence tagging, differentiating human-written tokens (0) from machine-generated tokens (1). An additional encoder trained on a blend of raw and annotated texts further refined sequence tagging. The average change point positions predicted by both encoders served as the final boundary estimation.

Team USTC-BUPT$_{\text{STC:3}}$ (Guo et al., 2024) approached the task as a token classification challenge, opting to fine-tune a DeBERTa model enhanced by data augmentation techniques derived from the training set. They reported that DeBERTa-base outperformed other models, and explored the efficacy of sequence labeling (e.g., BIOS) in detecting boundaries within mixed texts. The potential of various layers, including CRF and Dropout, was also examined for their impact on system performance.

Team RKadiyala$_{\text{STC:6}}$ (Kadiyala, 2024) focused on fine-tuning various encoder-based models appended with a Conditional Random Field (CRF) layer, noting that Deberta-V3 yielded the best results on the development set.

Team DeepPavlov$_{\text{STC:7}}$ (Voznyuk and Konovalov, 2024) fine-tuned the Deberta-v3 model using a specially prepared dataset with augmented texts, created by modifying prefixes and suffixes of original texts. They emphasized the importance of augmented data quality in achieving a mean absolute error (MAE) of 15.20903.

Team RUG-5$_{\text{STC:17}}$ (Darwinkel et al., 2024) utilized an augmented Longformer model, incorporating extra features into the output state of each token to enrich them with contextual information. This approach aimed at improving token-level classification by leveraging linguistic-stylistic features beyond simple PLM optimization.

Team TueCICL$_{\text{STC:22}}$ (Stuhlinger and Winkler, 2024) experimented with character-level LSTMs and LSTMs using pretrained Word2Vec embeddings, demonstrating that smaller models could compete with transformer models in the boundary detection task.

Team jelarson$_{\text{STC:25}}$ (Larson and Tyers, 2024) explored rule-based methods and linear regression techniques, identifying specific patterns in the training data that could inform better data collection practices, such as ensuring a more randomized and unbiased dataset.

Team TueSents$_{\text{STC:26}}$ (Pickard and Do, 2024) extracted textual features at the sentence level using tools like SpaCy and trained a lightweight BiLSTM model for boundary prediction, achieving an accuracy of 0.7 and MAE of less than 0.5 on the development set.

Team MasonTigers$_{\text{STC:27}}$ (Puspo et al., 2024) combined TF-IDF, PPMI, and RoBERTa features with linear regression and Elastic Net, culminating in an ensemble approach based on a weighted development set.

Team Unibuc-NLP$_{\text{STC:28}}$ (Marchitan et al., 2024) framed the task as a token classification problem, merging character-level features (extracted via CNN) and word embeddings within a BiLSTM model, further exploring the addition of CRF for enhanced performance.

Author Index